UEP 50

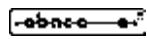
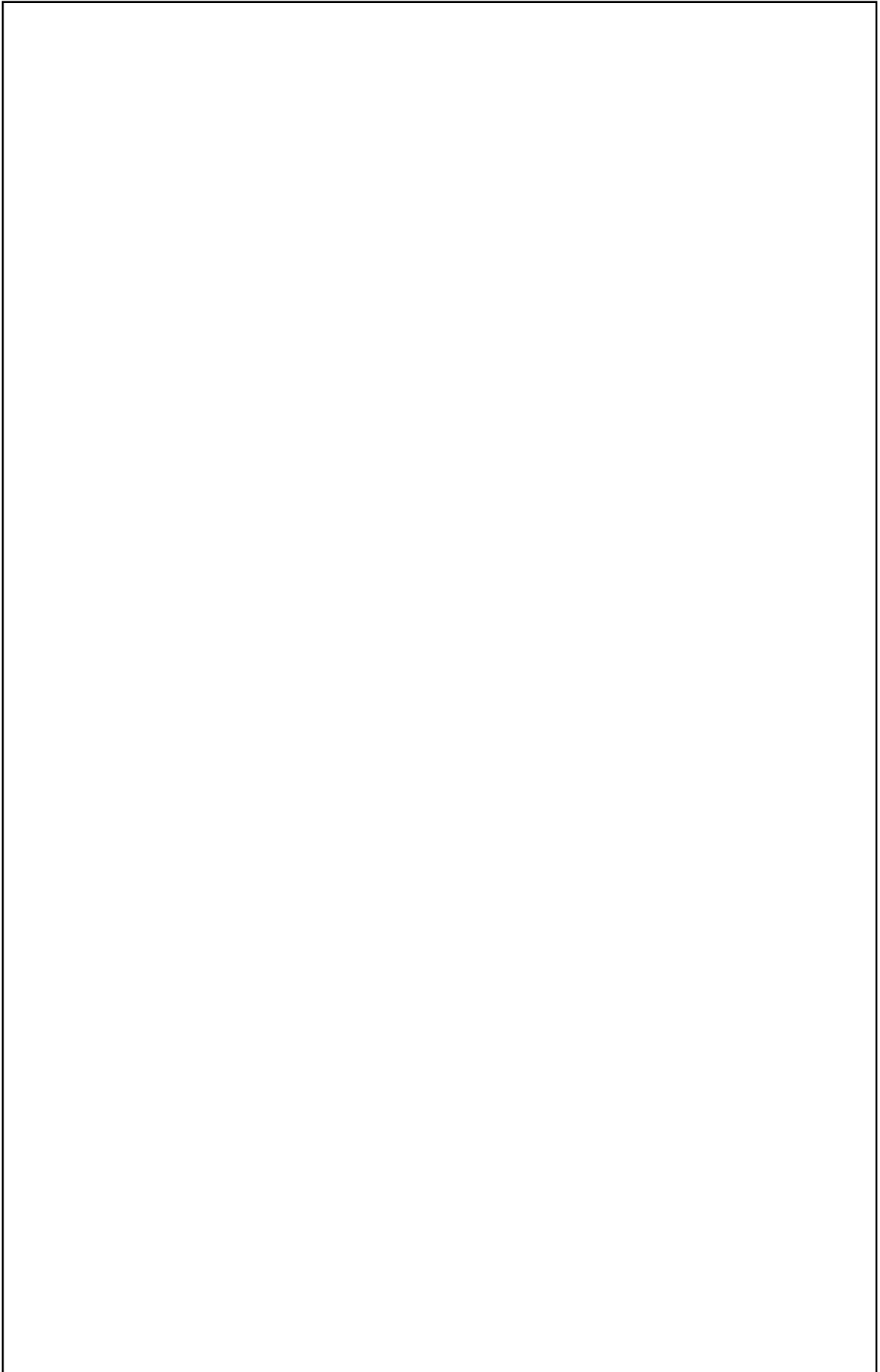
Universal Eprom Programmer - 48 pins, USB

MANUALE UTENTE

USER MANUAL

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it

http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661

UEP 50 Rel. 3.00 *Ed. 19 Ottobre 2012*

UEP 50

Universal Eprom Programmer - 48 pins, USB

MANUALE UTENTE

UEP 50 è un programmatore compatto, veloce e potente in grado di supportare tutti i dispositivi programmabili, presenti sul mercato.
Tramite il connettore **ISP** è inoltre in grado di programmare direttamente un dispositivo già installato sulla scheda dell'utente.
L' interfaccia di comunicazione e l'esecuzione autonoma delle operazioni da parte dell'elettronica interna, ottimizzano al massimo sia i tempi di trasferimento dati, che quelli complessivi di programmazione.
La sua concezione permette di aggiungere facilmente nuovi dispositivi e di fornire un prezzo molto competitivo insieme ad una elevata affidabilità. Indubbiamente offre uno dei migliori rapporti prezzo/prestazioni tra i programmatori della sua classe.

grifo®
ITALIAN TECHNOLOGY
Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661

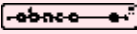
UEP 50 Rel. 3.00 *Ed. 19 Ottobre 2012*

IMPORTANTE

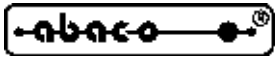
Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo®** non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo®**.
Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:

Attenzione: Pericolo generico

Attenzione: Pericolo di alta tensione

Attenzione: Dispositivo sensibile alle cariche elettrostatiche

MARCHI REGISTRATI

INDICE GENERALE

INTRODUZIONE 1

INFORMAZIONI PRELIMINARI 3
CONVENZIONI E TERMINOLOGIA 3
CONVENZIONI 3
TERMINOLOGIA 3

CARATTERISTICHE GENERALI 4

CARATTERISTICHE TECNICHE 8
MATERIALE RICEVUTO 8
GENERALI 8
DAC, ZOCCOLO E PINDRIVER 8
INTERFACCIA ISP 10
TEMPI DI PROGRAMMAZIONE 10
ELEMENTI DELL'UEP 50 11
OPERAZIONI SUI DISPOSITIVI 12
NORMALI 12
SICUREZZA 12
SPECIALI 12
OPERAZIONI SUL BUFFER 12
TIPI DI FILES SUPPORTATI 13
FAMIGLIE DISPOSITIVI GESTITI 13
PROGRAMMAZIONE PARALLELA SU ZOCCOLO ZIF 13
PROGRAMMAZIONE SERIALE CON INTERFACCIA ISP 14
TESTER DI CIRCUITI INTEGRATI 15
CONFIGURAZIONE MINIMA DEL PC 15
CONFIGURAZIONE CONSIGLIATA DEL PC 15

INSTALLAZIONE ED USO 16
INSTALLAZIONE PROGRAMMA DI CONTROLLO 16
PREPARAZIONE PROGRAMMATORE 18
COLLEGAMENTO USB: INSTALLAZIONE DRIVER 18
COLLEGAMENTO LPT 20
AGGIORNAMENTO PROGRAMMA DI CONTROLLO 21
PRECAUZIONI D'USO 21
LEDS 22
MULTIPROGRAMMAZIONE 22
TEST DIAGNOSTICO (SELFTEST) 23
BOCCOLE DI TERRA E PROTEZIONE ESD 23
PROGRAMMAZIONE ISP 24
PRECAUZIONI PER PROGRAMMAZIONE ISP 24
USO PROGRAMMAZIONE ISP 25
CONNETTORE ISP 26

DESCRIZIONE SOFTWARE 30
ESECUZIONE DEL PROGRAMMA DI CONTROLLO 30
SCHERMATA PRINCIPALE 31
TASTI SPECIALI 32
COMANDI DEL PROGRAMMA DI CONTROLLO 33
FILE 33
BUFFER 36
COMPONENTE 39
PROGRAMMATORE 47
OPZIONI 50
AIUTO 52
CARATTERISTICHE SOFTWARE AGGIUNTIVE 54

COME INIZIARE 55

TERMINI DELLA GARANZIA 57

ARTICOLI AGGIUNTIVI 57

RISOLUZIONE DEI PROBLEMI 58
PROLEMI DI LETTURA O PROGRAMMAZIONE 58
ERRORI DI COMUNICAZIONE 59
DISPOSITIVI NON SUPPORTATI 60

APPENDICE A: INDICE ANALITICO A-1

INDICE DELLE FIGURE

FIGURA 1: VISTA COMPLESSIVA UEP 50 5
FIGURA 2: CONTENUTO DELLA CONFEZIONE 9
FIGURA 3: TEMPI DI PROGRAMMAZIONE 10
FIGURA 4: POSIZIONE LEDS, CONNETTORI, PULSANTE, ECC. 11
FIGURA 5: FINESTRA UTILIZZO MINI CD 16
FIGURA 6: RIASSUNTO DELLE SELEZIONI D'INSTALLAZIONE 17
FIGURA 7: INSTALLAZIONE DRIVER USB (1 DI 3) 18
FIGURA 8: INSTALLAZIONE DRIVER USB (2 DI 3) 19
FIGURA 9: INSTALLAZIONE DRIVER USB (3 DI 3) 19
FIGURA 10: DESCRIZIONE LEDS 22
FIGURA 11: POD PER SELFTEST PLUS 23
FIGURA 12: POD PER SELFTEST ISP 23
FIGURA 13: COLLEGAMENTO PER PROGRAMMAZIONE ISP 24
FIGURA 14: CONNETTORE ISP 26
FIGURA 15: COLLEGAMENTO SEGNALI ISP SU PROGRAMMATORE 27
FIGURA 16: SEGNALI ISP 28
FIGURA 17: FINESTRA PRINCIPALE PG4UW 31

INTRODUZIONE

L’uso di questi dispositivi è rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.

Questo prodotto non è un **componente di sicurezza** così come definito dalla direttiva **98-37/CE**.

Alcuni componenti del programmatore sono sensibili ai fenomeni ESD: il personale che lo utilizza e maneggia i dispositivi usati è invitato a prendere tutte le precauzioni necessarie per evitare i possibili danni che potrebbero derivare dalle cariche elettorostatiche.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all’uso competente e sicuro dei prodotti. Esse sono il frutto di un’elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall’ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare i nostri siti www.grifo.it o www.grifo.com o direttamente l’Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell’uso di tali dispositivi, é conveniente che l’utente - PRIMA DI COMINCIARE AD OPERARE - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all’indice generale e all’indice analitico, posti rispettivamente all’inizio ed alla fine del manuale.

Le informazioni contenute in questo manuale sono aggiornate alla data di rilascio del medesimo, ma i nostri prodotti subiscono continui aggiornamenti e miglioramenti. Si prega di consultare l'help in linea per eventuali cambiamenti dell'ultimo minuto.

INFORMAZIONI PRELIMINARI

UEP 50 è un programmatore universale ed un tester di circuiti integrati logici, dotato di 48 potenti pin driver, con molteplici capacità e funzionalità. Questa struttura permette di aggiungere facilmente il supporto a nuovi dispositivi, di mantenere il costo molto competitivo rispetto ai concorrenti della stessa classe, di ottenere un eccellente rapporto prezzo/prestazioni, ed allo stesso tempo un'elevata affidabilità della programmazione.
La presenza dell'interfaccia ISP e delle interfaccie di comunicazione seriale (USB) allargano i suoi campi di utilizzo e la sua praticità d'uso.

CONVENZIONI E TERMINOLOGIA

In questo manuale vengono usate alcune convenzioni ed alcuni termini specifici:

CONVENZIONI

I riferimenti alle funzioni del programma di controllo sono in grassetto come ad esempio **CARICA**, **FILE**, **DISPOSITIVO**, ecc.
I riferimenti ai pulsanti di controllo sono racchiusi tra le parentesi angolari <>, ad esempio <**F1**>.

TERMINOLOGIA

Termine	Descrizione
Dispositivo target	Un qualunque tipo di dispositivo o circuito integrato programmabile.
Zoccolo ZIF	Zoccolo a forza di inserzione nulla usato per inserire il dispositivo target nel programmatore.
PC	Un Personal Computer necessario per consentire all'utente di interagire con il programmatore.
Buffer	Area di memoria, o del disco rigido, utilizzata per memorizzare temporeneamente dei dati.
Porta USB	Interfaccia seriale ad alta velocità del PC, generalmente usata per connettere periferiche esterne come stampanti, dischi esterni, scanner, modem, fotocamere digitali, ecc.
ISP	In System Programming, ovvero la capacità di programmare un dispositivo target senza toglierlo dalla scheda in cui si trova, e quindi anche quando è già saldato. La scheda utente deve essere compatibile con questa caratteristica.
Formato HEX	Formato per file dati che può essere letto con un normale editor di testo; ad esempio il byte 5AH viene memorizzato con i caratteri '5' ed 'A' corrispondenti ad i bytes 35H e 41H. Ogni linea di questo tipo di file (denominata record) contiene un'indirizzo iniziale un identificatore, i bytes di dati, ed infine un checksum di sicurezza.
DIL	Formato del contenitore dei componenti a doppia fila di piedini per montaggio tradizionale (Dual In Line).
PG4UW	Programma di controllo eseguito dal PC che gestisce il programmatore.

CARATTERISTICHE GENERALI

Questo capitolo contiene una descrizione discorsiva e concisa del programmatore, incluso un sommario delle sue caratteristiche tecniche ed operazionali. Per ottenere informazioni dettagliate sulle caratteristiche tecniche e d'uso del programmatore si consiglia di esaminare i capitoli successivi.

UEP 50 è il primo membro dell'innovativa generazione di programmatori con comunicazione USB nato dalla decennale esperienza **grifo®**, da utilizzarsi in abbinamento ad un PC provvisto di sistema operativo Windows e progettato per rispondere alle rigorose esigenze dei tecnici sia nei laboratori di sviluppo che in produzione.

UEP 50 supporta **tutti i componenti elettronici** al silicio diffusi sull'odierno e futuro mercato, lasciando all'utente la libertà di scegliere il componente che meglio soddisfa le sue esigenze di progettazione. La disponibilità della programmazione ISP consente di gestire direttamente i componenti compatibili, anche quando sono già montati sulla scheda finale.

UEP 50 non é solo un programmatore, ma anche un **tester** di circuiti integrati logici TTL/CMOS e di memorie. In aggiunta può generare sequenze di combinazioni di test stabilite dall'utente consentendo la verifica di componenti particolari e/o speciali.

Ha un prezzo molto competitivo ma è allo stesso tempo un programmatore affidabile grazie al suo eccellente progetto hardware. In questo modo offre il **miglior rapporto prezzo/prestazioni** nella sua gamma.

UEP 50 ha dei **tempi di programmazione ridotti**, grazie all'hardware basato su FPGA ed all'esecuzione degli algoritmi di programmazione direttamente all'interno del programmatore. E' sempre veloce come tutti i suoi concorrenti ma per molti dispositivi é decisamente più veloce di loro. In questo modo, quando é usato in produzione, accade che il programmatore spesso attende l'operatore, non altre condizioni circostanti.

Il programmatore si interfaccia a qualsiasi personal computer IBM o compatibili, con processore Pentium, sia nelle versioni da tavolo che portatili, dotati di una porta USB (1.1, 2.0).

Così il programmatore può sfruttare l'alta velocità sia della porta seriale USB.

UEP 50 é dotato di una boccola per il collegamento dei **bracciali antistatici** in modo da assicurare la protezione contro i fenomeni ESD, e di una boccola per il collegamento a **terra**.

UEP 50 è dotato di 48 potenti **pindrivers** TTL che generano livelli H/L/pull_up/pull_down e forniscono la capacità di lettura su ogni pin dello zoccolo. I nostri pindriver sono tecnologicamente avanzati e permettono di ottenere programmazioni di alta qualità, ad alta velocità, senza sovraelongazioni di tensione o rimbalzi di massa, per tutti i dispositivi supportati. Inoltre sono in grado di lavorare fino a 1,8 V cosicchè sarete già in grado di programmare la totalità dei dispositivi a bassa tensione attualmente sul mercato.

Il programmatore esegue il **test di corretto inserimento** del dispositivo e di **buon contatto elettrico** prima di iniziare le operazioni. Tali capacità, integrate dal controllo del byte di identificazione (signature) e dalle protezioni contro le sovracorrenti sui pin, aiutano a prevenire danneggiamenti ai componenti dovuti ad errori dell'operatore.

Tramite alcuni accessori ed il programma di gestione forniti, l'**UEP 50** é in grado di eseguire un **test diagnostico**, che verifica automaticamente i pindrivers, i livelli di tensione, le temporizzazioni e la comunicazione col PC.
Delle apposite **circuiterie di protezione interne** assicurano che il programmatore ed i dispositivi usati non si danneggino a causa di errori esterni. Tutti gli ingressi dell'**UEP 50**, inclusi lo zoccolo ZIF, l'interfaccia ISP, i collegamenti al PC e l'ingresso di alimentazione, sono protetti contro le scariche ESD fino a 15 KV.

UEP 50 effettua **verifiche** di programmazione **ai livelli limite** della tensione di alimentazione, il che, ovviamente, migliora l'affidabilità e garantisce una lunga conservazione dei dati.

FIGURA 1: VISTA COMPLESSIVA UEP 50

UEP 50 è pilotato da un programma di controllo **facile da usare**, con menù a tendina, tasti di scelta rapida ed aiuto in linea. Si può selezionare un dispositivo target mediante la classe a cui appartiene, oppure dal nome del produttore o semplicemente digitando una parte della sua sigla. Oltre alle normali operazioni eseguibili sui dispositivi (lettura, verifica cancellazione, programmazione, verifica di programmazione, cancellazione), sono state implementate alcune **funzioni di controllo** aggiuntive (verifica di inserzione, controllo del byte di identificazione) ed alcune funzioni speciali (autoincremento, modalità di produzione, partenza immediata all'inserzione del componente). Il programma di controllo permette la manipolazione dei dati del dispositivo temporaneamente salvati dentro un apposito **buffer**, di trasferire il buffer da e su un file su disco, esegue la rilevazione automatica del tipo di file e la eventuale conversione.

La ricca funzione di **autoincremento** dell'**UEP 50** permette di assegnare numeri di serie individuali ad ogni dispositivo programmato. Tale funzione semplicemente incremente un numero di serie nel buffer ogni volta che un nuovo dispositivo viene programmato. Inoltre, la funzione permette all'operatore di prelevare i numeri di serie e/o le informazioni di identificazione da un file.

Il programma di gestione fornisce molte informazioni aggiuntive specifiche del dispositivo scelto. Ad esempio, potete trovare i **disegni con quote** di tutti i contenitori disponibili per il componente ed il significato di tutti i prefissi e suffissi che compongono la **sigla** del dispositivo.
Il software fornisce anche tutte le informazioni per la programmazione ISP: la descrizione del connettore ISP a seconda del dispositivo scelto, la circuiteria d'interfacciamento consigliata, le modalità d'uso ed altre informazioni necessarie.

La caratteristica del **controllo remoto** consente al programma di controllo di essere gestito da altre applicazioni, usando degli appositi file di comando .BAT oppure delle librerie .DLL. Nella normale fornitura sono sono presenti sia la documentazione specifica che esempi (in C, PASCAL, VBASIC, .NET, ecc.) per questa interessante possibilità.

I **file JAM** secondo lo standard JESD-71, sono generati dagli ambienti di sviluppo del costruttore e definiscono la totalità delle informazioni necessarie per la programmazione su zoccolo ZIF o tramite ISP (IEEE 1149.1 del Joint Test Action Group=interfaccia JTAG). Il programma di controllo dell'**UEP 49+** é in grado di interpretare questi file JAM, tramite un suo apposito esecutore definito JAM Player.

I **file .VME** sono invece interpretati dall'esecutore VME Player. Tali file sono una variante binaria compressa dei file .SVF e contengono operazioni ad alto livello secondo lo standard IEEE 1149.1. Tali file sono generati dagli ambienti di sviluppo del costruttore e definiscono la totalità delle informazioni necessarie per la programmazione su zoccolo ZIF o tramite ISP (IEEE 1149.1 del Joint Test Action Group=interfaccia JTAG).

Molti dispositivi possono essere programmati e testati tramite l'**interfaccia JTAG** con le due associazioni appena descritte: ISP-JAM oppure ISP-VME.

Diversi programmatori **UEP 50** possono essere collegati allo stesso PC (tramite le sue porte USB) ottenendo un potente sistema di **multiprogrammazione** che gestisce tanti componenti quanti sono i programmatori collegati, senza diminuire la velocità di programmazione. E' importante ricordare che in questo caso la multiprogrammazione é concorrente ed ogni programmatore lavora indipendentemente dagli altri, ottenendo la possibilità di gestire contemporaneamente sia componenti uguali che diversi.

L'alta velocità, la multiprogrammazione ed i segnali di stato verso sistemi esterni dell'**UEP 50** gli consentono di essere usato con profitto anche nei reparti di **produzione** infatti risulta preferibile (per prestazioni e flessibilità) anche a numerosi programmatori multizoccolo (gang programmer).

E' importante ricordare che nella maggioranza dei casi l'aggiunta di nuovi dispositivi richiede solo l'**aggiornamento del software** di controllo, infatti l'**UEP 50** é un vero programmatore universale. Con il servizio di assistenza immediata si può ottenere l'aggiunta del nuovo componente alla lista dei dispositivi, in pochi giorni.

Sono inoltre disponibili diversi convertitori dal formato DIL a PLCC, SOIC, PSOP, SSOP, TSOP, TSSOP, TQFP, QFN (MLF), SDIP, BGA, LAP, CSP ed altri **contenitori**. Il programma di gestione riconosce il formato in base alla sigla completa del dispositivo e, quando necessario, informa con il nome ed il codice d'ordine del convertitore.

.

CARATTERISTICHE TECNICHE

MATERIALE RICEVUTO

- Programmatore **UEP 50**.
- Cavo USB per collegare il progammatore al PC.
- Cavo ISP (vedere appositi paragrafi).
- Cavo per alimentazione da rete.
- Scheda per auto diagnosi dello zoccolo ZIF del programmatore.
- Scheda per auto diagnosi del connettore ISP del programmatore.
- Copertura anti polvere per zoccolo ZIF.
- Mini CD con il programma di controllo e altre di utilità (compreso il presente manuame utente).
- Stampa del certificato di calibrazione post collaudo.
- Confezione di cartone.

Tali componenti possono essere verificati nella figura 2 che li mostra tutti ad eccezione degli ultimi due.

GENERALI

- Tensione di alimentazione da 110 a 240 Vac, 50÷60 Hz.
- Consumo potenza massimo di 20 W in piena attività, circa 2W in sospensione.
- Porta di comunicazione USB compatibile 2.0 ed 1.1, fino a 480 Mbit/s.
- Dotato di potente microprocessore ed FPGA interni per la gestione degli stati macchina.
- Boccola a banana per bracciale antistatico dell'operatore.
- Boccola a banana per connessione a terra.
- Protezioni contro sovratensioni e scariche ESD su alimentazione e porte di comunicazione.
- Dimensioni 197 x 140 x 56 mm (A x L x P).
- Contenitore metallico con piedi in gomma.
- Peso 1100 g (solo programmatore); 1550 g (intera confezione).
- Temperatura da 5 a 40 °C.
- Umidità da 20 a 80%, senza condensa.

DAC, ZOCCOLO E PINDRIVER

- 3 D/A converter per VCCP e VPP1 e VPP2, con tempi di discesa e salita controllabili.
- VCCP nel range da 0 a 8 V; 1 A.
- VPP1 e VPP2 nel rangeda da 0 a 26 V; 1 A.
- Capacità di test diagnostico autonomo (Selftest).
- Driver TTL basato su FPGA per generare H, L, CLK, pull-up e pull-down su tutti i pin
- Driver per pin: 48 driver universali in grado di:
    - collegare ogni pin a VCCP/VPP1/VPP2;
    - collegare perfettamente ogni pin a GND;
    - collegare ogni pin al driver TTL.
- Uscita del driver analogico per pin selezionabile da 1,8 V fino a 26 V.
- Circuiterie di protezione con limitazione della corrente: interrompono l'erogazione in caso di sovracorrenti o cadute dell'alimentazione.
- Zoccolo a 48 pins DIL ZIF (Zero Insertion Force) accetta direttamente dispositivi da 300 a 600 mil.
- Protezioni contro sovratensioni e scariche ESD sui pin dello zoccolo, secondo standard IEC1000-4-2: 15 KV in aria, 8 KV su contatto.
- Continuità: ogni pin viene verificato prima di iniziare la programmazione.

FIGURA 2: CONTENUTO DELLA CONFEZIONE

INTERFACCIA ISP

- Un connettore a scatolino, maschio, 20 vie, con chiave per inserzione corretta.
- Sei TTL driver per ISP generano H, L, CLK, pull-up e pull-down, livello analogico di uscita H selezionabile da 1,8 V fino a 5 V per gestire tutti i dispositivi, inclusi quelli a bassa tensione.
- Una tensione VCCP per ISP (2÷7 V, 100 mA).
- La VCCP per ISP può sia erogare che assorbire corrente ed il suo livello viene monitorato.
- Una tensione VPP per ISP (2÷25 V, 50 mA).
- Tensione di alimentazione per dispositivo target (2÷6 V, 250 mA).
- Protezioni contro sovratensioni e scariche ESD sui pin dello zoccolo, secondo standard IEC1000-4-2: 15 KV in aria, 8 KV su contatto.
- Due segnali di uscita digitali che riportano lo stato dell'ultima operazione eseguita: LEDOK e LEDERROR (livello minimo segnale attivo= 1,8 V).
- Un segnale d'ingresso digitale per prosecuzione della prossima operazione, equivalente al pulsante YES! (livello massimo segnale attivo= 0,8 V).

TEMPI DI PROGRAMMAZIONE

Questi tempi dipendono fortemente dal collegamento effettuato ma non dalla configurazione del PC. Nella seguente tabella vengono riportati alcuni tempi di programmazione ottenuti con un PC *Pentium IV; 2,4GHz; USB 2.0 HS; Windows XP*.

DISPOSITIVO, TIPO DIMENSIONI	OPERAZIONE	MODO	TEMPO
M50FW080, Flash parallela 100000Hx8=8 MBit=1 MByte	programmazione e verifica	su zoccolo ZIF	22 sec
MX28F640C3BT, Flash parallela 400000Hx16=64 MBit=8 MByte	programmazione e verifica	su zoccolo ZIF	57 sec
K9F1G08U0M, Flash NAND par. 8400000Hx8=1 GBit=128 MByte	programmazione e verifica	su zoccolo ZIF	239 sec
AT45D081, Flash seriale 108000Hx8=8 MBit=1 MByte	programmazione e verifica	su zoccolo ZIF	36 sec
AT89C51RD2, microcontrollore 10000Hx8=512 KBit=64 KByte	programmazione e verifica	su zoccolo ZIF	15 sec
PIC18LF452, microcontrollore 4000Hx16=256 KBit=32 KByte	programmazione e verifica	su zoccolo ZIF	4 sec

FIGURA 3: TEMPI DI PROGRAMMAZIONE

ELEMENTI DELL'UEP 50

Il programmatore é composto dai seguenti elementi direttamente utilizzabili dall'utente:

1) Pulsante YES!
2) LED di accensione/sospensione (POWER)
3) Connettore di alimentazione
4) Zoccolo ZIF da 48 pin
5) LED per operazione in corso (BUSY)
6) Connettore USB per comunicazione con PC
7) Connettore ISP (vedere paragrafo PROGRAMMAZIONE ISP)
8) Connettore LPT (Opzione)
9) LED per operazione avvenuta correttamente (GOOD)
10) Boccola a banana per collegamento a terra
11) LED per operazione avvenuta con errori (ERROR)
12) Interruttore accensione e spegnimento
13) Boccola a banana per bracciale antistatico

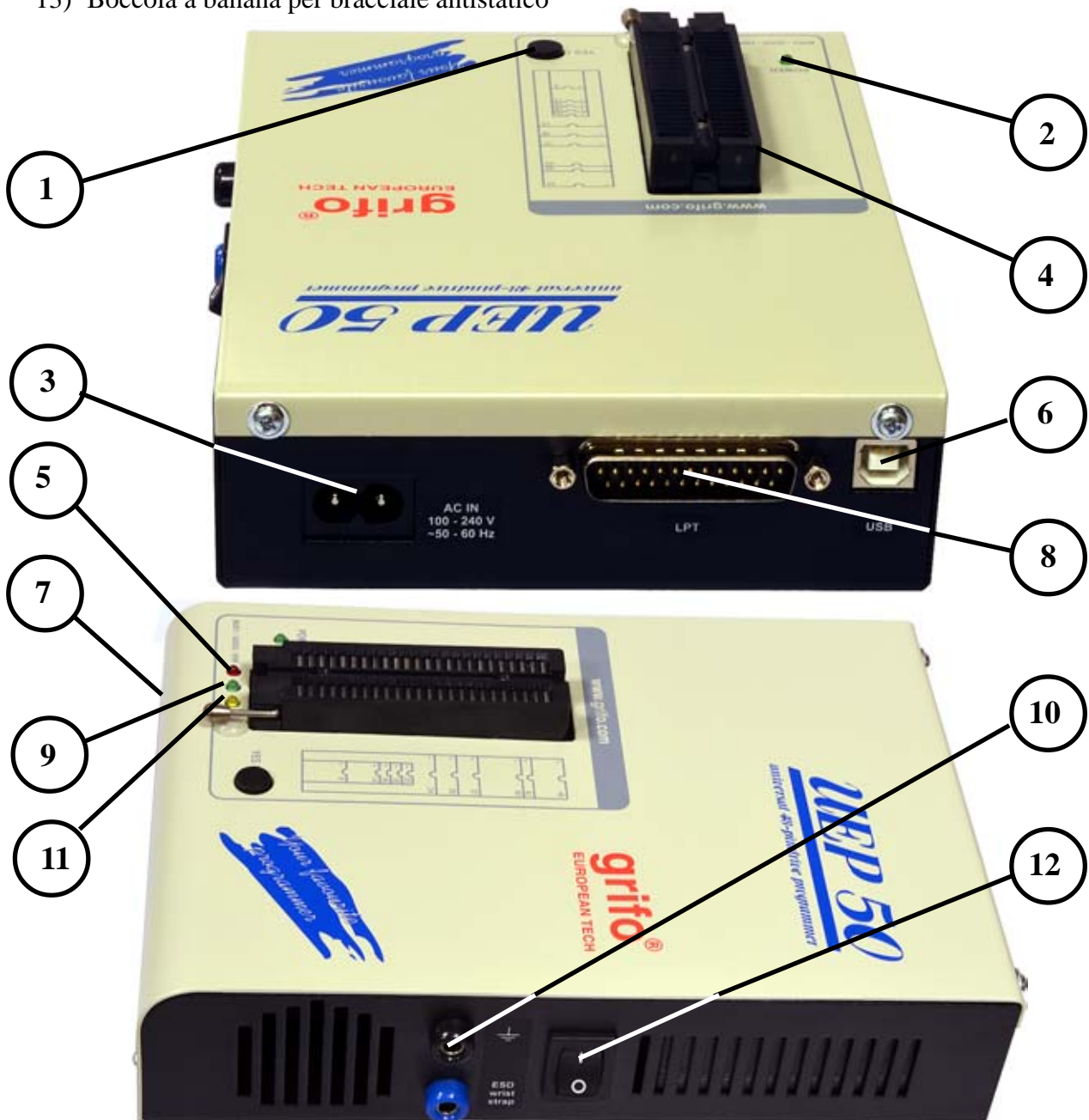



FIGURA 4: POSIZIONE LEDS, CONNETTORI, PULSANTE, ECC.

OPERAZIONI SUI DISPOSITIVI

Il programmatore **UEP 50** in abbinamento al suo programma di controllo su PC, consente di effettuare i seguenti gruppi di operazioni. Qualora il dispositivo scelto abbia diverse memorie interne, si ricorda che le seguenti operazioni sono disponibili per tutti tali tipi di memorie.

NORMALI

- Selezione intelligente del dispositivo per tipo, produttore o frazione della sua sigla.
- Selezione automatica di EPROM, Flash EPROM e microcontrollori in base al byte di identificazione interno del componente (Signature).
- Verifica dispositivo vuoto, lettura e di corretta programmazione.
- Programmazione.
- Cancellazione.
- Test bit illegali.
- Checksum.

SICUREZZA

- Verifica di inserzione corretta del dispositivo.
- Controllo della bontà dei contatti elettrici.
- Controllo del byte di identificazione (Signature).
- Gestione fusibili di protezione.

SPECIALI

- Statistiche.
- Incremento automatico del numero di serie con diversi tipi di variazione: semplice incremento, da file, generato da utente.
- Conto alla rovescia dei dispositivi programmati.
- Gestione fusibili di configurazione.
- Interprete dei file JAM e del linguaggio di programmazione STAPL, secondo lo standard di programmazione e test JEDEC JESD-71.
- Interprete dei file SVF (Serial Vector Format) e del relativo formato binario compressoVME.
- Modalità di produzione con partenza automatica dell'operazione immediatamente dopo l'inserzione del dispositivo.

OPERAZIONI SUL BUFFER

In questo paragrafo vengono elencate le operazioni che il programmatore può svolgere sul buffer che contiene i dati letti o da scrivere sul dispositivo usato.

- Esamina/modifica, trova/sostituisci.
- Riempi/copia, sposta, scambia byte, spezza word o doppie word.
- Checksum (in diversi formati standard).
- Stampa.
- Salva su e carica da disco.
- Niente tempi di scaricamento né caricamento in quanto il programmatore è controllato da PC.
- Identificazione automatica del tipo di file.

TIPI DI FILES SUPPORTATI

Il programma di controllo dell'**UEP 50** é in grado di gestire automaticamente i seguenti formati di file:

- Binario (senza dati di formattazione aggiuntivi).
- HEX: Intel, Intel esteso.
- Motorola S-record.
- MOS.
- Exormax.
- Tektronix.
- ASCII-space-HEX.
- ASCII HEX.
- JEDEC (ver. 3.0.A), generato dai pricipali programmi come ABEL, CUPL, PALASM, WARP, TANGO PLD, OrCAD PLD, PLD Designer ISDATA ecc.
- JAM (JEDEC STAPL Format), JBC (Jam STAPL Byte Code), STAPL (STAPL File) secondo lo standard JEDEC JESD-71.
- VME (file ISP VME, versioni VME2.0 eVME3.0).

FAMIGLIE DISPOSITIVI GESTITI

Di seguito vengono descritte le principali famiglie di componenti gestibili con l'**UEP 50**. Per un elenco completo, aggiornato e dettagliato sui componenti programmabili si consiglia di usare il programma di gestione e le sue modalità di ricerca.

PROGRAMMAZIONE PARALLELA SU ZOCCOLO ZIF

- EPROM: NMOS/CMOS, serie 2708*, 27xxx e 27Cxxx, con larghezza bus dati 8/16 bit, pieno supporto per modelli a bassa tensione LV.
- EEPROM: serie NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx, con larghezza bus dati 8/16 bit, pieno supporto per modelli a bassa tensione LV.
- Flash EPROM: serie 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx, da 256 KBit fino a 1 GBit, con larghezza bus dati 8/16 bit e pieno supporto per modelli a bassa tensione LV.
- EEPROM seriali: serie 17Cxxx, 24Cxxx, 24Fxxx, 25Cxxx, 25Fxxx, 25Pxxx, 45Dxxx, 59Cxxx, 85xxx, 93Cxxx, NVM3060, MDAxxx con pieno supporto per modelli a bassa tensione LV.
- EEPROM per configurazioni: serie XCFxxx, XC17xxxx, XC18Vxxx, EPCxxx, AT17xxx, 37LVxx, compresi i modelli a bassa tensione LV.
- EEPROM con interfaccia 1-Wire: serie DS1xxx, DS2xxx.
- PROM: prodotte da AMD, Harris, National, Philips, Signetics, Tesla, TI.
- NV RAM: serie Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x.
- PLD Altera: serie MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX7000AE, MAX II.
- PLD Lattice: serie ispGAL22V10x, ispLSI1xxx, ispLSI1xxxEA, ispLSI2xxx, ispLSI2xxxA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, LC4xxxB/C/V/ZC, M4-xx/xx, M4A3-xx/xx, M4A5-xx/xx, M4LV-xx/xx.

- PLD Xilinx: serie XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II.
- PLD di altri costruttori: SPLD/CPLD prodotte da AMI, Atmel, AMD-Vantis, Gould, Cypress, ICT, Lattice, NS, Philips, STM, VLSI, TI.
- Microcontrollori famiglia 51: serie 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx, di tutti i produttori, serie LPC di Philips e Nexpedia.
- Microcontrollori Intel 196: serie 87C196 KB/KC/KD/KT/KR/...
- Microcontrollori Atmel AVR: serie ATtiny, AT90Sxxx, ATmega.
- Microcontrollori Cypress: serie CY7Cxxxxx,, CY8Cxxxxx.
- Microcontrollori ELAN: serie EM78Pxxx.
- Microcontrollori MDT: serie 1xxx and 2xxx.
- Microcontrollori Microchip PIC®: serie PIC10xxx, PIC12Cxxx, PIC14xxx, PIC16Cxxx, PIC17Cxxx, PIC18xxx, PIC24xxx, dsPIC.
- Microcontrollori Motorola (Freescale): serie 68HC05, 68HC08, 68HC11, HCS08, HCS12.
- Microcontrollori Myson: serie MTV2xx, 3xx, 4xx e 5xx.
- Microcontrollori National: serie COP8xxx.
- Microcontrollori NEC: serie uPD78Fxxx.
- Microcontrollori Novatek: serie NT68xxx.
- Microcontrollori Renesas: serie M3xxx, HD64Fxxxx.
- Microcontrollori Scenix (Ubicom): serie SXxxx.
- Microcontrollori SGS-Thomson: serie ST6xx, ST7xx, ST10xx, STR7xx.
- Microcontrollori TI: serie MSP430 e MSC121x.
- Microcontrollori ZILOG: serie Z86/Z89xxx e Z8xxx.
- Altri microcontrollori prodotti da: EM Microelectronic, Fujitsu, Goal Semiconductor, Hitachi, Holtek, Princeton, Macronix, Winbond, Infineon (Siemens), Samsung, Toshiba, ecc.

PROGRAMMAZIONE SERIALE CON INTERFACCIA ISP

- EEPROM seriali: modelli con interfaccia I2C, MicroWire, SPI ed altri modelli come la serie KEELOQ, le Data Flash seriali, le memorie di configurazione PLD.
- Microcontrollori Atmel: serie AT89Sxxx, AT90Sxxxx, ATtiny, ATmega.
- Microcontrollori Cypress: serie CY8C2xxxx.
- Microcontrollori Elan: serie EM6xxx, EM78Pxxx.
- Microcontrollori EM Microelectronic: serie a 4 ed 8 bit.
- Microcontrollori Microchip PIC®: serie PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, PIC24xxx, dsPIC.
- Microcontrollori Motorola/Freescale: serie HC11, HC908 (sia a 5 fili che tutti i fili), HCS08, HCS12.
- Microcontrollori NEC: serie uPD7xxx.
- Microcontrollori Philips, Nexpedia: serie LPCxxx, LPC2xxx, 89xxx.
- Microcontrollori Scenix (Ubicom): serie SXxxx.
- Microcontrollori TI: serie MSP430 (sia JTAG che BS), MSC12xxx.
- PLD Lattice: serie ispGAL22xV10x, ispLSI1xxxEA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, M4-xx/xx, M4LV-xx/xx, M4A3-xx/xx, M4A5-xx/xx, LC4xxxB/C/V/ZC.
- PLD varie (anche tramite JAM player con supporto JTAG):
  Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 9000, MAX II
  Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II.

TESTER DI CIRCUITI INTEGRATI

- Porte TTL tipo: 54,74 S/LS/ALS/H/HC/HCT.
- Porte CMOS tipo: 4000, 4500.
- RAM statiche: 6116÷624000.
- Combinazioni di test definibili da utente.

NOTE:
(*) Dispositivo obsoleto, gestibile con moduli aggiuntivi.

CONFIGURAZIONE MINIMA DEL PC

Di seguito sono riportati i requisiti minimi del PC che viene collegato ed usato con il programmatore; tali requisiti sono relativi alle attuali caratteristiche del programma di controllo e possono quindi variare al variare di quest'ultimo. Per ottenere informazioni aggiornate cercare nell'aiuto in linea del **PG4UW**.

- Pentium III.
- 128 MB di RAM libera.
- Hard disk con 60 MB di spazio libero.
- Un lettore CD.
- Una portaUSB (1.1 o superiore) oppure una porta LPT (ECP/EPP).
- Monitor a colori con 800 x 600 punti.
- Mouse e tastiera.
- Sistema operativo: MS Windows 2000/XP.

CONFIGURAZIONE CONSIGLIATA DEL PC

Lo spazio libero consigliato sull'hard disk dipende dal dispositivo con cui si deve lavorare. In generale é conveniente disporre di almeno 60 MB + dimensione dispositivo.

- Pentium IV.
- 512 MB di RAM libera.
- Hard disk con 150 MB di spazio libero.
- Un lettore CD.
- Una portaUSB 2.0 High Speed.
- Monitor a colori con 1024 x 768 punti.
- Mouse e tastiera.
- Sistema operativo: MS Windows XP.

INSTALLAZIONE ED USO

In questo capitolo vengono riportate le informazioni principali relative alla messa in funzione del programmatore, complete di alcuni suggerimenti per l'utente che ne semplificano e ne assicurano l'uso in ogni condizione operativa. Si consiglia quindi di eseguire le operazioni descritte nei paragrafi di questo capitolo nell'ordine riportato e quindi di non collegare il programmatore al PC prima di aver installato il programma di controllo.
Si ricorda inoltre di consultare anche i capitoli successivi che approfondiscono la descrizione del programma di gestione e riportano i passi per eseguire velocemente una prima programmazione.

INSTALLAZIONE PROGRAMMA DI CONTROLLO

1) Selezionare il PC da utilizzare in abbinamento al programmatore che soddisfi i requisiti descritti nel precedente capitolo.
2) Verificare che sul PC non sia già stato installato il programma di controllo PG4UW ed in questo caso provvedere a disinstallarlo. Per fare questo, dal menu di avvio scegliere "Programmi", poi “Grifo(r)” e infine “Uninstall Pg4uw”. Seguire le istruzioni a video fino a completare la disinstallazione.
3) Nella confezione viene fornito un mini CD contenente il programma di controllo, i manuali utente dei programmatori ed il programma Acrobat Reader dedicato alla lettura degli stessi manuali.
   Inserite il mini CD nel lettore del PC, un programma di presentazione partirà automaticamente visualizzando la seguente finestra. Qualora la finestra non appaia, provvedere a lanciare manualmente il programma "SETUP.EXE" presente sullo stesso CD.

FIGURA 5: FINESTRA UTILIZZO MINI CD

4) Premere il pulsante "Software instalation PROGRAMMERS" ed attendere la partenza del programma di installazione.

5) Selezionare la lingua "Italiano" da usare durante l'installazione e premere il pulsante "Ok".
6) Leggere la seguente finestra informativa e provvedere a chiudere ogni altra applicazione eventualmente in esecuzione e premere il pulsante "Avanti".
7) Selezionare la cartella sull'hard disk del PC in cui installare il programma di controllo e confermare con il pulsante "Avanti". A questo punto il programma avvisa dell'eventuale esistenza della cartella scelta e l'utente può decidere se utilizzarla comunque oppure no.
8) Selezionare la cartella con il collegamento di lancio del programma da aggiungere al menù di "Avvio" di Windows e premere il pulsante "Avanti".
9) Selezionare quali e quante icone di avvio rapido aggiungere sul PC (come ad esempio quella sulla scrivania, quella sulla barra di Windows, ecc.).
   Inoltre scegliere se il supporto al controllo della Multiprogrammazione deve essere installato, come descritto nel successivo paragrafo MULTIPROGRAMMAZIONE.
   A questo punto premere il pulsante "Avanti".
10) Controllare che nella finestra riassuntiva rappresentata tutte le selezioni effettuate siano corrette e quindi confermare con il pulsante "Installa".

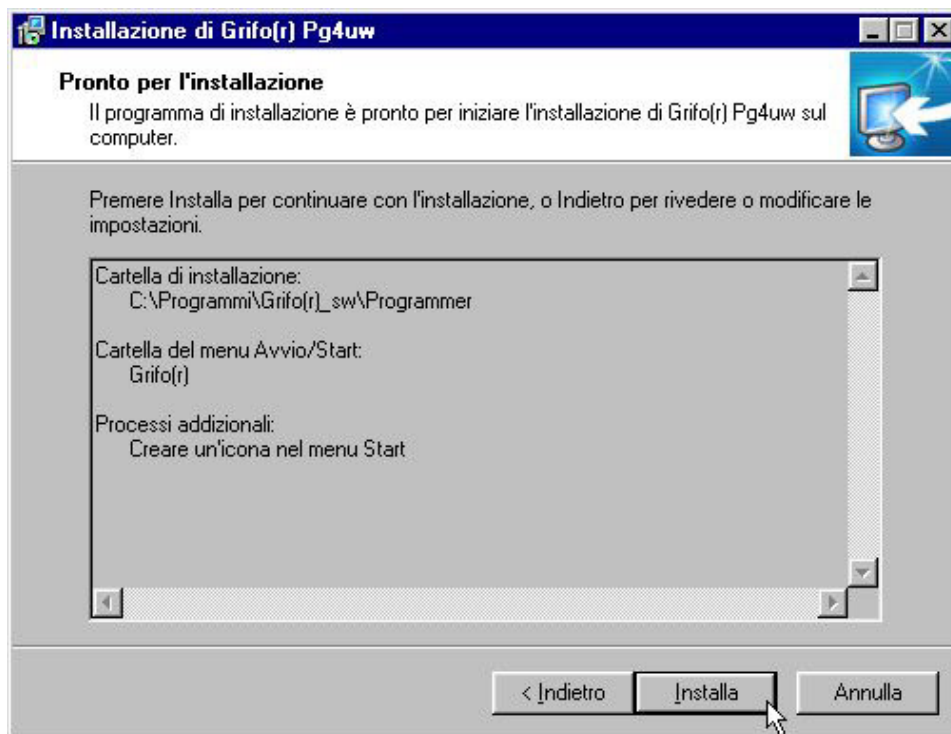



FIGURA 6: RIASSUNTO DELLE SELEZIONI D'INSTALLAZIONE

11) Attendere la fine dell'installazione evidenziata da un'apposita barra a scorrimento.
12) A questo punto a seconda del PC e del sistema operativo usato, può comparire una finestra che richiede il riavvio del PC: selezionare "Si, riavvia il sistema adesso". In ogni caso premere il pulsante "Fine" con cui il programma d'installazione si chiude definitivamente e ritorna alla videata di figura 5.
13) In questa fase tutto il software necessario al programmatore é già stato copiato sull'hard disk del PC. Per la consultazione del manuale utente sul PC serve il programma Acrobat Reader che, se non già presente sul PC, può essere convenientemente installato premendo il pulsante "Install Adobe Acrobat Reader".
14) Uscire dal programma di presentazione premendo il pulsante "Exit" e poi "Close".
15) Per praticità d'uso é comunque conveniente copiare anche il manuale utente, in formato PDF, dalla cartella \Documentation\ sul mini CD, all'hard disk del PC.

PREPARAZIONE PROGRAMMATORE

16) Verificare che tutti gli elementi del programmatore (descritti nel precedente paragrafo MATERIALE RICEVUTO) siano presenti nella confezione.
17) Collegare il cavo di alimentazione da rete ad una presa elettrica a 230 Vac ed al programmatore.
18) Posizionare l'interruttore di accensione sulla posizione I e verificare che immediatamente dopo si accenda il LED POWER ad alta luminosità, poi si accendano e si spengano in sequenza i tre LEDs BUSY, GOOD, ERROR ed infine che rimanga acceso solo il LED POWER a bassa luminosità. Questa è la normale sequenza di accensione con PC non collegato.

A questo punto le operazioni da effettuare si differenziano a seconda che il programmatore sia collegato al PC con l'interfaccia seriale USB o quella parallela LPT. I due paragrafi seguenti illustrano entrambe queste modalità:

COLLEGAMENTO USB: INSTALLAZIONE DRIVER

Come per ogni periferica del PC collegata tramite la porta USB, anche per l'**UEP 50** il sistema operativo deve disporre degli appositi driver software che ne consentano la gestione. Tali driver vengono copiati durante l'installazione del programma di controllo, quindi per evitare inutili complicazioni si consiglia prima di effettuare l'installazione descritta nei punti 4 -18 e solo dopo collegare il programmatore.
Le istruzioni seguenti valogono per Windows XP, ma la procedura di installazione per gli altri sistemi operativi Windows è molto simile.

19) Collegare il programmatore al PC tramite il cavo USB fornito nella confezione.
20) A questo punto si avvia automaticamente l'installazione guidata di un nuovo hardware che si compone di alcune fasi consecutive, come descritto nei seguenti punti. Si ricorda che durante tali fasi il mini CD inserito al punto 3 si deve trovare ancora nel lettore del PC.
21) Nella prima fase bisogna scegliere di "Installare il software automaticamente" come illustrato nella seguente figura e proseguire con il pulsante "Avanti".

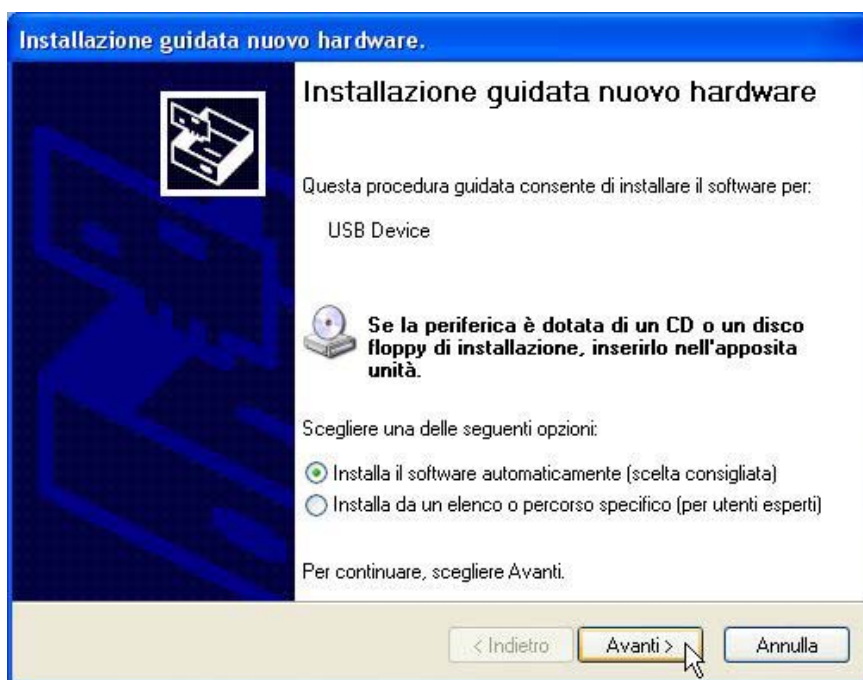



FIGURA 7: INSTALLAZIONE DRIVER USB (1 DI 3)

22) Successivamente il PC cercherà i driver migliori per il nuovo hardware collegato sui dischi del PC, trovandoli sul mini CD fornito. A questo punto bisogna continuare l'installazione dei driver anche se questi non rispettano la compatibilità Windows, premendo il pulsante "Continua".

FIGURA 8: INSTALLAZIONE DRIVER USB (2 DI 3)

23) Al termine dell'installazione e registrazione dei driver controllare che la periferica funzioni correttamente verificando che la finestra rappresentata coincida con quella sottostante e quindi chiuderla con il pulsante "Fine".

FIGURA 9: INSTALLAZIONE DRIVER USB (3 DI 3)

24) A questo punto l'installazione dei driver USB é terminata ed il programmatore é pronto ad operare. Si ricorda che in caso di successivi collegamenti o riaccensioni del programmatore i driver USB saranno già presenti e quindi non richiesti, ammesso che si continui ad usare la stessa porta USB del PC. Se invece tale porta viene cambiata si dovranno ripetere i passi 20÷23.

COLLEGAMENTO LPT (OPZIONALE)

In alternativa al collegamento seriale USB l'**UEP 50** può essere collegato tramite l'interfaccia parallela LPT del PC. Per chiarezza si ricorda che sul PC tale interfaccia corrisponde a quella dedicata al comando della stampante, secondo lo standard Centronics o IEEE1284.
Anche per questa connessione al fine di evitare inutili complicazioni si consiglia prima di effettuare l'installazione descritta nei punti 4÷18 e dopo eseguire i seguenti passi.

19) Spegnere PC e programmatore.
20) Procuratevi un cavo per la porta parallela del PC ovvero una prolunga D a 25 vie con collegamenti 1:1, dotato di un connettore maschio, un connettore femmina e schermatura. In caso di necessità tale cavo può essere acquistato anche da rivenditori di accessori per computer.
21) Inserire il cavo di comunicazione in una porta parallela libera del vostro PC, rimuovendo l'eventuale cavo per stampante già presente. Infine collegare l'altra estremità del cavo all'**UEP50** e stringete le viti di entrambi i connettori. Questo è molto importante per evitare malfunzionamenti randomici.
    Anche se l'operazione di sostituzione dei cavi della stampante e del programmatore quando si usa l'uno o l'altro dispositivo, é molto scomoda si deve comunque ricordare che non è possibile collegare il programmatore tramite un data switch, sia meccanico che elettronico. In questo caso potete vantaggiosamente installare una scheda supplementare di I/O sul PC e dedicare la nuova porta, ad esempio LPT2, all'**UEP 50**, e mantenere la stampante sempre collegata alla porta LPT1.
22) Riattivare il PC.
23) Riaccendere il programmatore verificando che si ripeta la sequenza di accensione dei LEDs, già descritta al punto 18
24) A questo punto l'**UEP 50** è pronto ad operare.

! ATTENZIONE !

Se non volete spegnere e riaccendere il PC mentre aggiungete o togliete il programmatore, procedete come segue:
- Quando connettete il programmatore al PC: **PRIMA** inserite il cavo di comunicazione LPT, **POI** fornire l'alimentazione al programmatore.
- Quando scollegate il programmatore dal PC: **PRIMA** togliere l'alimentazione al programmatore, **POI** il cavo di comunicazione LPT.

Dal punto di vista del programmatore la sequenza di collegamento e scollegamento è irrilevante, i circuiti interni di protezione lo tengono al sicuro, **ma ciò non vale per il vostro PC**.

Qualora la comunicazione tra programmatore e PC, descritta nel capitolo successivo DESCRIZIONE SOFTWARE, non avvenga correttamente ricontrollare i punti sopra elencati e, se il problema persiste, consultare il capitolo RISOLUZIONE DEI PROBLEMI.

AGGIORNAMENTO PROGRAMMA DI CONTROLLO

Le aziende che costruiscono componenti aggiungono continuamente nuovi dispositivi, con nuovi contenitori, sviluppano nuove tecnologie in modo da soddisfare le nuove richieste del mercato.
Inoltre numerosi componenti subiscono variazioni durante il loro ciclo di vita (le cosiddette revisioni di maschera) che correggono il dispositivo o lo mantengono tecnicamente aggiornato. Queste variazioni implicano una modifica negli algoritmi di programmazione che definiscono appunto le modalità di gestione del componente. Usare un algoritmo di programmazione aggiornato si traduce anche in un migliore mantenimento dei dati programmati, in un ridotto numero di componenti difettosi durante la scrittura ed in un ridotto tempo di programmazione.
Infine errare é umano e nel programma di controllo ci possono essere errori sfuggiti a tutti i controlli.

Per queste ragioni la **grifo®** sviluppa periodicamente un nuovo programma di controllo che risolve tutti i problemi elencati. Ad esempio dalla prima all'ultima versione di un anno vengono mediamente aggiunti 500 nuovi dispositivi, decine di algoritmi e corretti alcuni errori. Per ottenere il massimo dalle capacità del programmatore consigliamo di usare sempre la versione più aggiornata di PG4UW.EXE. Per ottenere tale versione si deve:

- Disinstallare il programma già presente selezionando l'apposito collegamento in “Programmi” | “Grifo(r)” | “Uninstall Pg4uw”. Seguire le istruzioni a video fino a completare la disinstallazione. Se durante la disinstallazione compaiono richieste di eliminazione di driver confermare facendo attenzione a verificare che questi riguardano solo il programma PG4UW.
- Scaricare gratuitamente dai siti internet: **http://www.grifo.com**
  http://www.grifo.it
  da cui si ottiene il file "PG4UWARC.EXE”.
- Lanciare il programma scaricato che coincide con il programma di installazione della nuova versione.
- A questo punto è possibile installare il nuovo programma effettuando i passi da 5 a 23 riportati nei precedenti paragrafi.

Si consiglia di verificare l'uscita di nuove versioni almeno una volta al mese oppure quando lo stesso programma di controllo lo consiglia alla partenza.

PRECAUZIONI D'USO

L'elettronica incorporata nell'**UEP 50** protegge il programmatore stesso ed il dispositivo in uso contro numerose condizioni anomale di utilizzo. Comunque non è possibile garantire la salvaguardia dei due sistemi in tutte le possibili combinazioni d'uso e per questo l'utente deve prendere le seguenti precauzioni:

- il dispositivo target da programmare può essere danneggiato se l'utente effettua selezioni errate od inserisce parametri sbagliati nel programma di controllo;
- il dispositivo target può essere distrutto da interruzioni forzate del programma di controllo (reset o spegnimento del PC) o dalla rimozione della connessione fisica col programmatore. Se il componente non si danneggia avrà comunque un contenuto inaffidabile;
- non togliete il dispositivo target dallo zoccolo ZIF durante l'accesso da parte dell'**UEP 50**, ovvero quando il LED BUSY è acceso;
- un posizionamento errato del componente sullo zoccolo può provocare un danneggiamento od una distruzione del componente;
- le cariche elettrostatiche possono generare risultati imprevedibili su tutti i dispositivi elettronici;
- rispettare tutte le precauzioni riportate nel paragrafo PROGRAMMAZIONE ISP.

LEDS

UEP 50 è fornito di 4 LEDs per visualizzare lo stato dell'alimentazione e lo stato interno del programmatore. Fate riferimento alla figura 4 per localizzare i LEDs.

NOME LED	COLORE	DESCRIZIONE
POWER	Verde	Segnala lo stato di alimentazione del programmatore. Può assumere due stati: alta luminosità in caso di accensione completa; bassa luminosità in caso di basso consumo (sospensione).
BUSY	Rosso	Quando acceso indica che il programmatore sta accedendo al dispositivo target in lettura o scrittura. Non toccare tale dispositivo quando il LED è acceso.
GOOD	Verde	Quando acceso indica che l'ultima operazione eseguita si è conclusa con successo.
ERROR	Giallo	Quando acceso indica che si è verificata una condizione di errore durante l'ultima operazione eseguita.

FIGURA 10: DESCRIZIONE LEDS

NOTE:
Se l'**UEP 50** non viene usato, il programmatore si pone in sospensione (sleep) e riduce il consumo a meno di 2 Watt; tale consumo può essere azzerato agendo sull'interruttore di accensione e spegnimento presente a lato del programmatore.
A seguito della fase di accensione i tre LEDs BUSY, GOOD, ERROR vengono accesi e spenti in sequenza in modo da consentire all'utente di verificarne la funzionalità.

MULTIPROGRAMMAZIONE

L'**UEP 50** prevede la multiprogrammazione ovvero la gestione contemporanea di diversi programmatori, collegati in USB, tramite un solo PC. I seguenti punti illustrano brevemente i passi fondamentali da svolgere per usare questa modalità:

- Iinstallare il supporto alla multiprogrammazione (si veda punto 9 dell'INSTALLAZIONE PROGRAMMA DI CONTROLLO).
- Effettuare il collegamento dei programmatori e la relativa installazione dei driver USB.
- Eseguire l'apposito programma PG4UWMC.EXE e selezionare il programmatore usato.
- Caricare i progetti di configurazione per ogni programmatore collegato.
- Eseguire il programma di controllo PG4UW per ogni programmatore collegato e configurato.

La multiprogrammazione si rivolge soprattutto alle esigenze di produzione di numerosi componenti uguali e/o diversi e normalmente é usata da due persone: il tecnico esperto che prepara i file di progetto e quello inesperto che invece si limita ad usare il programmatore già predisposto.
Per dettagli e chiarimenti leggere con attenzione l'apposita documentazione in linea del programma PG4UWMC che coordina la multiprogrammazione stessa.

TEST DIAGNOSTICO (SELFTEST)

Se ritenete che il programmatore non reagisca secondo le vostre aspettative, potete lanciare i test diagnostici che fanno uso di appositi POD diagnostici (per zoccolo ZIF e connettore ISP) inclusi nella confezione. Per avere prestazioni ottimali vi suggeriamo di effettuare comunque i test diagnostici circa ogni 6 mesi, in modo da anticipare eventuali anomalie di funzionamento che possono avere effetti indesiderati sui componenti da programmare.
Per informazioni sulle modalità di esecuzione dei test diagnostici si rimanda agli appositi paragrafi PROGRAMMATORE | SELFTEST, PROGRAMMATORE | SELFTEST PLUS... e PROGRAMMATORE | SELFTEST CONNETTORE ISP... nel capitolo DESCRIZIONE SOFTWARE ed alle relative pagine dell'aiuto in linea.
Si ricorda che che il primo di questi comandi deve essere eseguito con lo **zoccolo ZIF vuoto, altrimenti il programmatore potrebbe essere danneggiato**. Il secondo invece deve essere eseguito con il POD diagnostico per zoccolo ZIF inserito:

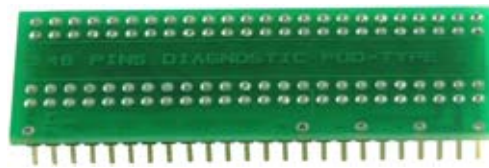



FIGURA 11: POD PER SELFTEST PLUS

Ed infine il terzo comando deve essere eseguito con il POD diagnostico per ISP inserito nello zoccolo ZIF e con il cavo ISP che collega il POD al connettore ISP del programmatore:

FIGURA 12: POD PER SELFTEST ISP

Al fine di avere un programmatore sempre in perfette condizioni di lavoro é consigliabile eseguire periodicamente anche una verifica di calibrazione, tramite l'apposito comando **Programmatore | Test calibrazione...** del programma di controllo. Tale verifica richiede un apposito POD di calibrazione da richiedere direttamente alla grifo®.

BOCCOLE DI TERRA E PROTEZIONE ESD

Sul lato destro del programmatore sono presente due boccole a banana con le seguenti funzioni:

Boccola di terra: può essere collegata a terra in modo da assicurare un corretto equilibrio dei potenziali tra i vari sistemi interconnessi, come ad esempio, quando si usa la programmazione ISP.

Boccola protezione ESD: può essere collegata alle protezioni contro le cariche elettrostatiche indossate dall'operatore che maneggia il programmatore ed i componenti usati. Tra queste protezioni si ricordano i bracciali, collane, cavigliere ed ogni altro accessorio con questa funzione.

Per individuare facilmente le boccole sul programmatore si può fare riferimento alla figura 4.

PROGRAMMAZIONE ISP

La programmazione In System Programming permette di gestire diverse volte un dispositivo già saldato sulla scheda utente o comunque sull'applicativo finale. Utilizzando una semplice interfaccia, il programmatore ISP comunica in modalità seriale con il dispositivo, riprogrammandone la memoria non volatile interna.
Questo fatto elimina la necessità di rimuovere il dispositivo dalla scheda su cui è installato ogni volta che lo si deve ripogrammare risparmiando così tempo e denaro sia durante lo sviluppo in laboratorio sia per l'aggiornamento dei parametri o del software sul campo.
Inoltre l'interfaccia ISP può essere usata anche per leggere il contenuto di tutte le memorie interne del dispositivo, offrendo la possibilità di prelevare dalla scheda una serie di parametri, dati, configurazioni impostate durante la precedente installazione oppure raccolte dalla stessa scheda durante il suo funzionamento.

PRECAUZIONI PER PROGRAMMAZIONE ISP

Si raccomanda specialmente di rispettare le seguenti regole per evitare il danneggiamento del programmatore ISP e/o del PC e/o del dispositivo target e/o della scheda su cui è installato:

- Assicurare che ci sia un **unico punto comune di massa** tra la scheda su cui si trova il dispositivo target, il programmatore ISP ed il PC.
- Per chi utilizza portatili o altri PC non connessi ad un punto comune di terra: si effettui una connessione mediante un conduttore esterno verso il punto comune di massa (si usi, ad esempio, una delle **vaschette metalliche esterne** dei connettori **del PC**).
- Anche i dispositivi collegati alla scheda su cui avviene la programmazione devono essere collegati al punto comune di massa.

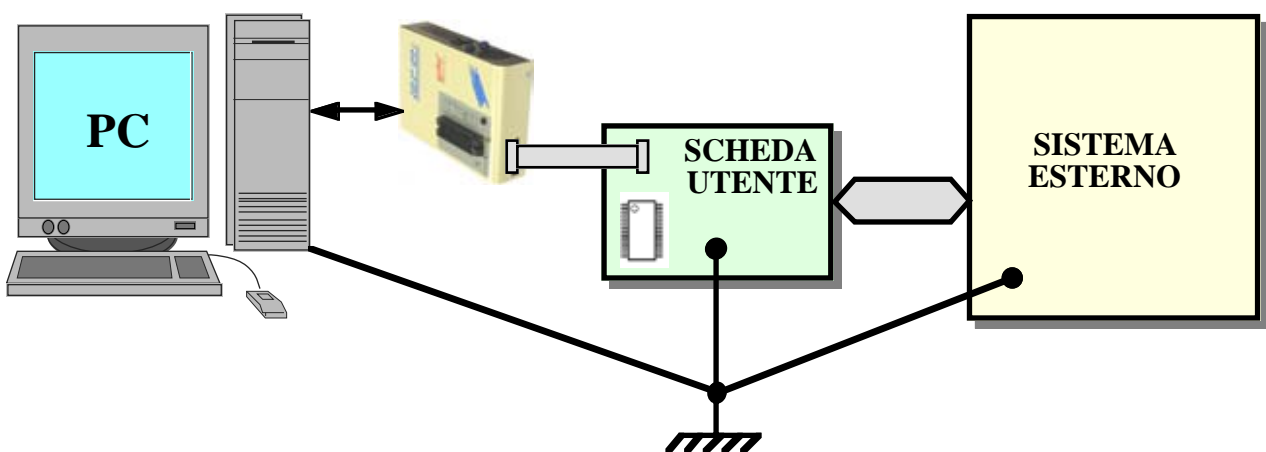



FIGURA 13: COLLEGAMENTO PER PROGRAMMAZIONE ISP

- Quando l'**UEP 50** viene usato come programmatore ISP, **non inserite dispositivi nello zoccolo ZIF**.
- Quando l'**UEP 50** viene usato come programmatore parallelo, **non collegare il connettore ISP**.
- Usate solo il **cavo ISP in dotazione**. Con altri cavi ISP (di diverso materiale, lunghezza, ecc.), la programmazione potrebbe essere non affidabile.

- **UEP 50** può alimentare il dispositivo target ma con una corrente limitata. A meno di opportune verifiche che confermino questa possibilità si consiglia di alimentare la scheda utente su cui é montato il dispositivo con una propria sorgente di alimentazione.
- La scheda su cui é montato il dispositivo target non può alimentare l'**UEP 50**: provvedere ad alimentare il programmatore con il proprio cavo ed accenderlo con l'interruttore laterale.
- **UEP 50** controlla la tensione di alimentazione del dispositivo target e ne verifica il valore all'inizio delle operazioni da effettuare. Se il valore misurato è diverso da quello atteso non verrà effettuata alcuna operazione sul dispositivo target.
- Durante la programmazione ISP avviene una connessione elettrica tra due dispositivi (il dispositivo target ed il programmatore ISP) che deve essere effettuata secondo le indicazioni del programma di controllo sul PC. Una connessione non effettuata propriamente potrebbe danneggiare entrambi i dispositivi.
- Se la scheda utente con il dispositivo target utilizza i segnali della programmazione ISP per altre circuiterie, è necessario separarli mediante opportuni resistori o jumpers. In altre parole tale scheda deve essere progettata in modo da prevedere la programmazione ISP (vedere descrizioni successive).

NOTA: ***Se il programmatore viene danneggiato durante una programmazione e non sono state rispettate le specifiche di cui sopra, la garanzia non è valida.***

USO PROGRAMMAZIONE ISP

Durante la programmazione ISP avviene una connessione elettrica tra due dispositivi (il dispositivo target da programmare ed il programmatore ISP). Una sequenza di connessione ed uso non effettuata correttamente potrebbe danneggiare entrambi i dispositivi. Dopo aver seguito tutte le precauzioni illustrate nel paragrafo precedente l'utente deve:

1) Spegnere sia dispositivo target che il programmatore. Per assicurare lo spegimento anche quando il programmatore é provvisto di interruttore d'accensione é preferibile liberare il connettore di alimentazione.
2) Collegare lo stesso punto comune di massa per tutti i dispositivi, ad esempio collegare tutte le masse insieme mediante un conduttore.
3) Inserire il cavo in dotazione al programmatore nel connettore ISP di quest'ultimo, alimentare il programmatore e lanciare il programma di controllo.
4) Selezionare il dispositivo target e le sue eventuali opzioni nel programma di controllo. Durante la scelta prestare attenzione alla presenza del suffisso ISP al termine della sigla del componente.
5) Iniziare l'operazioni richiesta (lettura, programmazione, ecc.).
6) Quando il programma di controllo indica di farlo, collegare il capo libero del cavo ISP alla scheda utente con il dispositivo target ed alimentarla. Una volta effettuate queste operazioni premere il pulsante **Ok** per proseguire.
7) A questo punto l'operazione iniziata al punto 5 prosegue fino a concludersi.
8) Quando il programma di controllo indica di farlo, rimuovere il connettore ISP dalla scheda utente e spegnerla.
9) Se dovete effettuare un'altra operazione sul dispositivo target, oppure ripetere l'operazione su un'altra scheda utente, tornate al punto 5.

CONNETTORE ISP

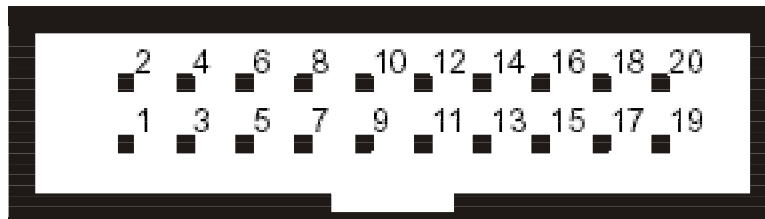
Il programmatore é provvisto di un connettore a scatolino, maschio, da 10 vie per la programmazione ISP. Tale connettore é facilmente accessibile in quanto situato sul lato inferiore del programmatore, come illustrato in figura 4, é provvisto di apposita chiave per evitare inserzioni errate e nella confezione é incluso anche un cavo a flat per il collegamento alla scheda utente.
La disposizione dei segnali sul connettore ISP dipende da quale dispositivo viene programmato e viene sempre specificata nella finestra **Informazioni componente** (che appare premendo la combinazione di tasti <**Ctrl+F1**>, dopo avere selezionato il dispositivo) ed in particolare nella pagina **Dettagli connessione ISP**.

La corrispondenza tra segnali ISP e pin del connettore può variare da dispositivo a dispositivo.
Per conoscerla, selezionare il dispositivo ISP desiderato nel programma di controllo (menu **Componente | Scelta componente**) e scegliere il comando **Componente | Informazioni componente** o premere <**Ctrl+F1**>.

FIGURA 14: CONNETTORE ISP

Una volta collegato il cavo ISP al connettore ISP la numerazione dei pin é comunque riconoscibile da un triangolo che contraddistingue il pin 1 sui entrambi i connettori femmina .

L'utente che intende usare la programmazione ISP deve progettare la sua scheda in modo che sia compatibile con questa modalità, ovvero che il programmatore possa essere direttamente collegato alla scheda tramite il suo connettore. Quando i segnali ISP del dispositivo target sono a loro volta usati per altre sezioni e funzioni della scheda é necessario isolarli. La sheda utente non deve alterare tali segnali durante la programmazione ISP.
Per ottenere questo risultato si devono seguire le indicazioni del dispositivo target, come ad esempio le note applicative, fornite dalla casa costruttrice. La grifo® rispetta completamente queste note per consentire una corretta programmazione in ogni condizione operativa. In alternativa si possono esaminare gli schemi elettrici riportati sempre nella pagina **Dettagli connessione ISP**.

In generale sulla scheda utente si dovrebbe prevedere quanto segue:

- Un diodo per proteggere i circuiti della scheda utente da tensioni elevate, quali possono essere quelle fornite per la programmazione ISP.
- Se la scheda utente non é alimentata a +5 Vdc, come protezione usare un diodo Zener con una tensione di soglia corretta all'alimentazione della scheda.

- La separazione dei segnali ISP collegati al programmatore da quelli usati a bordo scheda può essere convenientemente effettuata con resistenze in modo da non dover intervenire manualmente sulla scheda.
- Se la separazione tramite resistenze non é sufficiente si dovranno prevedere dei jumpers e l'utente dovrà muoverli sia prima che dopo la programmazione ISP.
- Se del connettore ISP si utilizzano i segnali JTAG fare riferimento alle normative di questo standard, ovvero IEEE 1149.
- Sul connettore ISP sono presenti due segnali in uscita che informano sullo stato dell'operazione svolta e possono essere usati da altri sistemi di produzione e/o test della scheda utente. Questi segnali sono previsti per il collegamento ad un LED e devono essere quindi collegati a segnali compatibili.
- Sul connettore ISP é presente un segnale d'ingresso che fa proseguire l'operazione in corso e può essere usato da altri sistemi di produzione e/o test della scheda utente. Questo segnale é previsto per il collegamento ad un pulsante normalmente aperto che chiude verso massa e deve essere quindi collegato ad un segnale compatibile.
- Ulteriori informazioni elettriche sui segnali ISP sono disponibili nel paragrafo INTERFACCIA ISP.

La circuiteria interna dell'**UEP 50** che gestisce i segnali dell'interfaccia ISP e JTAG, può essere paragonata alla circuiteria semplificata riportata nella seguente figura:

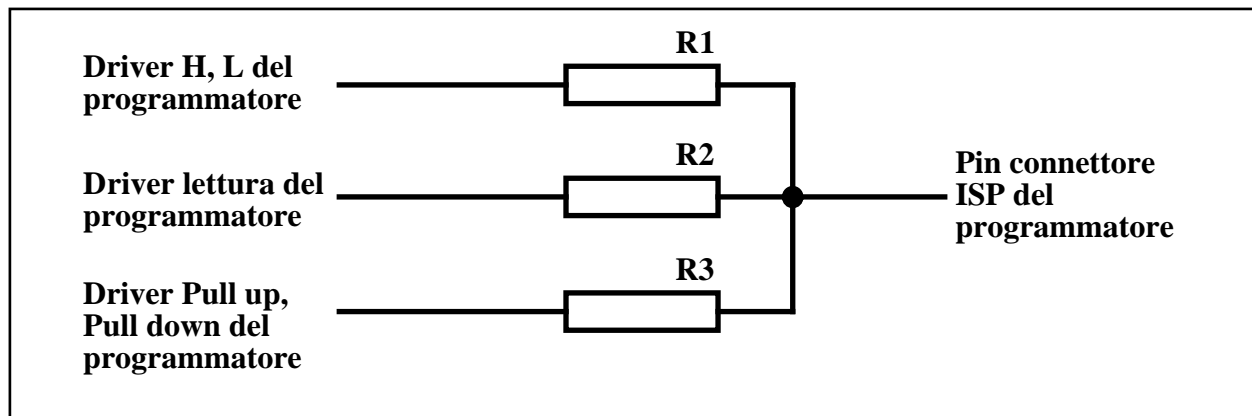



FIGURA 15: COLLEGAMENTO SEGNALI ISP SU PROGRAMMATORE

Qualora la documentazione trovata dei segnali usati nell'interfaccia ISP fosse insufficiente o mancante, l'utente può usare la seguente tabella che ne riporta i nomi ed una preve descrizione.

PIN	SEGNALE	SCOPO
PGM	ProGraM	Linea di abilitazione programmazione ISP a bassa (singola) tensione.
PGC	ProGram Clock	Clock per programmazione ISP, generato dal programmatore.
PGD	ProGram Data	Dati per programmazione ISP (In/Out).
MCLR	Master CLeaR	Segnale di reset attivo basso del dispositivo target. Per ottenere questa operazione il programmatore ISP deve poter pilotare il reset del dispositivo target.
Vpp	Programming Voltage	Tensione di programmazione per i dispositivi target che richiedono una doppia tensione.
SCK	Serial Clock	Clock di programmazione, generato dal programmatore ISP (master).
MOSI	Master Out Slave In	Linea di comunicazione dal programmatore ISP (master) al dispositivo target (slave).
MISO	Master In Slave Out	Linea di comunicazione dal dispositivo target (slave) al programmatore ISP (master).
RESET	Target Reset	Per abilitare la programmazione ISP il segnale RESET o /RESET del dispositivo target deve essere mantenuto attivo. Per ottenere questa operazione il programmatore ISP deve poter pilotare il reset del dispositivo target.
SDATA	Serial DATA	Dati per programmazione ISP (In/Out).
TDI	Test Data Input	Linea di comunicazione dal programmatore al dispositivo target tramite interfaccia JTAG.
TDO	Test Data Output	Linea di comunicazione dal dispositivo target al programmatore tramite interfaccia JTAG.
TCK	Test CLocK	Clock per programmazione tramite interfaccia JTAG.
YES!	YES	Ingresso digitale per pulsante di YES esterno (attivo quando collegato a massa), per avviare l'operazione tramite un sistema esterno.
LEDOK	LED OK	Uscita digitale per LED di OK esterno (attivo quando alto), per informare un sistema esterno del risultato corretto dell'ultima operazione svolta.
LED ERROR	LED ERROREe	Uscita digitale per LED di ERROR esterno (attivo quando alto), per informare un sistema esterno del risultato erratto dell'ultima operazione svolta.
Vdd	Power Supply voltage	Il dispositivo target ed il programmatore ISP devono essere alimentati ognuno dalla propria sorgente. Comunque il il programmatore ISP verifica la tensione di alimentazione, che quindi deve essere collegata.
GND (Vss)	Ground signal	I due sistemi (programmatore ISP e dispositivo target) devono avere la massa in comune.

FIGURA 16: SEGNALI ISP

La precedente tabella non riporta tutti i segnali ISP di tutti i dispositivi target programmabili con questa tecnica, ma solo quelli principali e maggiormente diffusi.

Per trovare la descrizione completa dei segnali ISP si rimanda l'utente alla documentazione tecnica del componente usato (data sheet e note applicative); una volta individuati tali segnali si può ricavare il loro collegamento con il connettore del programmatore ISP tramite la pagina **Dettagli connessione ISP** della finestra **Informazioni componente**, già descritta nelle pagine precedenti.

Ricordare che nel programma di controllo le finestre informative descritte variano al variare del componente scelto; per avere le giuste informazioni **é di fonfamentale importanza che l'utente abbia scelto correttamente il nome del dispositivo da usare, assicurando che al termine della sigla compaia sempre l'indicazione ISP**.

DESCRIZIONE SOFTWARE

Questo capitolo contiene una descrizione essenziale del programma di controllo **PG4UW.EXE**. Per completezza si ricorda che nei capitoli precedenti sono descritte le modalità d'installazione del programma e di collegamento del programmatore che devono essere naturalmente effettuate prima di usarlo. Inoltre nel capitolo successivo COME INIZIARE viene illustrata una breve sequenza di operazioni necessarie per effettuare una prima programmazione.
Di seguito vengono riportate alcune note generali del PG4UW:

- l'help in linea del programma viene aggiornato prima del manuale: l'utente deve quindi usare sempre questo comodo strumento;
- il programma è comune a tutti i programmatori **grifo®**;
- il programma opera correttamente in abbinamento ad un PC dotato dei requisiti minimi descritti nell'omonimo paragrafo;
- le operazioni in background sotto Windows sono prive di errori;
- il programma di controllo è garantito essere esente da qualunque virus al momento della consegna. Per incrementare la sicurezza i nostri programmi implementano un algoritmo speciale per rilevare possibili infezioni da virus.

ESECUZIONE DEL PROGRAMMA DI CONTROLLO

Per eseguire il PG4UW si può selezionare l'apposito collegamento nel menù di avvio di Windows (posizionato nella cartella scelta durante l'installazione) oppure con un doppio click su una delle icone aggiunte sempre durante la fase d'installazione.
Una volta partito PG4UW, verifica la propria integrità e se riconosciuta valida scandisce tutte le porte di comunicazione presenti sul PC e cerca un programmatore **grifo®** collegato ed alimentato. PG4UW è il programma comune a tutti i programmatori **grifo®** per cui cercherà tutti i modelli disponibili. Infine il programma mostra la finestra principale ed attende istruzioni.
Se il programma di controllo non riesce a comunicare col programmatore, appare sullo schermo un messaggio di errore che fornisce un codice di errore ed una descrizione delle possibili cause (programmatore scollegato, cattiva connessione, problemi di alimentazione, porta USB od LPT incompatibile, numero di serie del programmatore non valido, ecc.). Eliminate la causa dell'errore e premete un tasto qualunque.
Se la condizione di errore persiste, il programma entra in modalità demo e impedisce l'accesso al programmatore. Se non riuscite ad identificare la causa dell'errore, seguite le istruzioni nel capitolo RISOLUZIONE DEGLI ERRORI.
Inoltre, il programma di controllo verifica la comunicazione col programmatore prima di fare qualunque altra operazione col dispositivo da programmare.

SCHERMATA PRINCIPALE

Nella schermata principale sono sempre presenti le seguenti aree:

barra del **TITOLO**	mostra avvisi di copyright, nome e versione di PG4UW
barra dei **MENU**	elenco delle funzioni fondamentali
riquadro **STATISTICHE**	informazioni statistiche sull'attività del programmatore
riquadro **PROGRAMMATORE**	informazioni sullo stato interno di PG4UW
riquadro **INDIRIZZI (hex)**	organizzazione, dimensione, indirizzi di inizio e fine del dispositivo target, del buffer e del file
riquadro **COMPONENTE**	informazioni principali sull'attuale dispositivo target
barra di **AIUTO**	breve descrizione del comando selezionato.

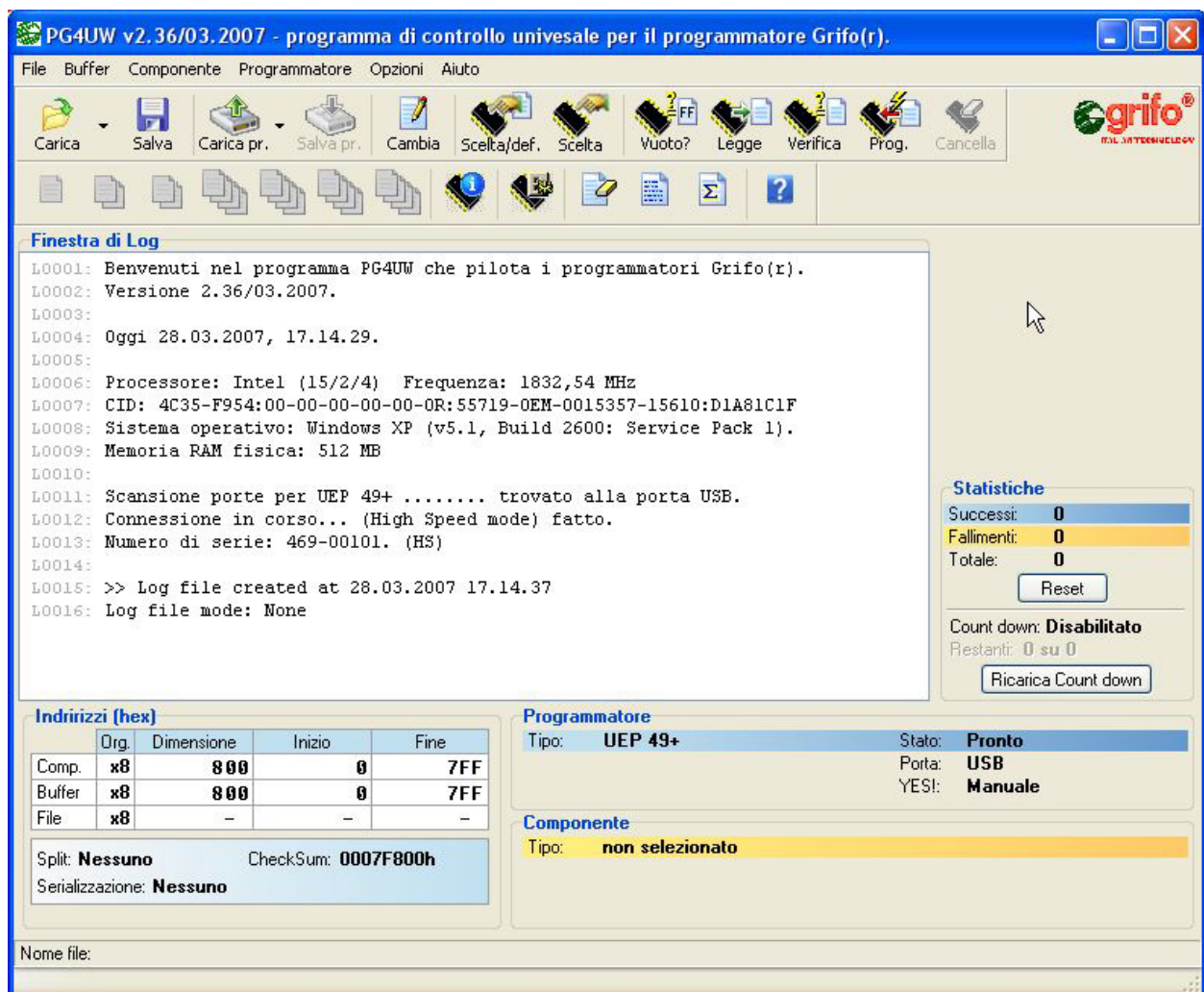



FIGURA 17: FINESTRA PRINCIPALE PG4UW

La selezione dei comandi avviene nel tipico modo delle interfacce grafiche utente - cioè muovendo il cursore e premendo <**ENTER**> oppure digitando la lettera evidenziata nel menu selezionato o, naturalmente, usando il mouse.

! ATTENZIONE !

Tutti i dati inseriti da tastiera sono in formato esadecimale, tranne i blocchi di caratteri ASCII nei menù **Buffer | Mostra/Cambia**.

TASTI SPECIALI

Il PG4UW prevede e gestisce anche delle associazioni tra tasti e i comandi di uso più frequente:

<**F1**>	Aiuto	Apre la finestra di aiuto in linea.
<**F2**>	Salva	Salva un file
<**F3**>	Carica	Carica un file nel buffer
<**F4**>	Mostra/Cambia	Esame/Modifica del buffer
<**F5**>	Scelta/default	Scelta del dispositivo target tra gli ultimi 10 dispositivi selezionati
<**Alt+F5**>	Scelta componente	Scelta del dispositivo target digitando il nome o quello del produttore
<**Ctrl+F5**>	Scelta per ID	Scelta del dispositivo target tramite suo codice interno di identificazione (signature)
<**F6**>	Verifica cancellazione	Verifica se il componente é completamente cancellato
<**F7**>	Lettura	Copia nel buffer il contenuto del dispositivo target
<**F8**>	Verifica	Confronta il contenuto del buffer con il contenuto del dispositivo target
<**F9**>	Programma	Programma il dispositivo target
<**Alt+Q**>	Esce senza salvare	Termina il PG4UW
<**Alt+X**>	Esce e salva	Termina il PG4UW e salva le impostazioni
<**Ctrl+F1**>	Informazioni componente	Mostra informazioni aggiuntive sul dispositivo target scelto
<**Ctrl+F2**>	Cancellazione	Riempie il buffer con un valore assegnato
<**Ctrl+Shift+F2**>	Riempimento casuale	Riempie il buffer con un valori casuali
<**Ctrl+F**>	Trova programmatore	Ricerca ogni programmatore collegato al PC
<**Shift+Ctrl+F**>	Ritrova programmatore	Ricerca il programmatore già collegato al PC
<**Shift+Alt+D**>	Lista componenti	Genera un elenco di tutti i dispositivi gestiti da tutti i programmatori

COMANDI DEL PROGRAMMA DI CONTROLLO

Le seguenti pagine contengono i comandi disponibili nel programma di controllo, raggruppati nei vari menù di cui il programma dispone. Accanto al nome del comando è presente, se esiste, la combinazione di tasti che lo richiama rapidamente e una breve descrizione del suo scopo e modalità d'uso. Si ricorda che l'aggiornamento più recente delle spiegazioni di questi comandi è sempre presente nell'aiuto in linea sensibile al contesto del programma di controllo, che quindi andrebbe consultato prima del presente manuale.

FILE

Viene usato per gestire i files sorgenti, esaminare le cartelle su disco, cambiare drive e modiicare gli indirizzi di inizio e di fine nel salvataggio e caricamento dei file di tipo **binario**, **MOTOROLA**, **MOS Technology**, **Intel (extended) HEX**, **Tektronix, ASCII space**, **JEDEC**, e **POF**.

File | Carica... <F3>

Analizza il formato del file e carica i dati nel buffer. Si può selezionare il tipo di file desiderato (**binario**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** e **POF**). Il programma di controllo memorizza l'ultimo tipo di file selezionato. Per rendere la memorizzazion persistente, salvare la configurazione con **Opzioni | Salva opzioni**.

Spuntando la casella **Automatic file format recognition** il programma riconosce automaticamente il formato. Se il formato non può essere riconosciuto, il file viene trattato come binario.

Togliendo lo spunto alla casella **Automatic file format recognition** il programma permette all'utente di specificare il tipo di file desiderato nella lista della casella di riepilogo **Selected file format**. Si può specificare l'indirizzo di inizio nel buffer con il formato binario. Tale indirizzo indica da quale posizione nel buffer saranno scritti i dati letti dal file.

Il tasto **<F3>** farà apparire questo menu da qualunque posizioni in qualunque momento.

File | Salva... <F2>

Salva su disco i dati contenuti nel buffer, creati, modificati o letti da un dispositivo. Si può selezionare il tipo di file desiderato (**binario**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** o **POF**).

Il tasto **<F2>** farà apparire questo menu da qualunque posizioni in qualunque momento.

File | Carica progetto...

Carica un file di progetto, che contiene la configurazione per i dispositivi da usare e la configurazione dell'interfaccia utente.

La finestra standard **Carica progetto** contiene un'altra finestra - **Descrizione progetto** - nella parte bassa. Quest'ultima mostra le informazioni contenute nel file attualmente selezionato nella finestra **Carica progetto**.

Le informazioni contenute in un progetto sono:
- nome e produttore del primo dispositivo selezionato nel progetto
- data e ora di creazione del progetto
- versione del programma con cui il progetto è stato creato
- descrizione testuale del progetto redatta dall'utente (si tratta di testo a piacere, di solito il nome degli autori ed alcune note)

File | Salva progetto...

Quest'opzione serve a salvare il file di progetto, il quale contiene i dati di configurazione del sipostitivo salvati. I dati di un progetto salvato possono essere richiamati in ogni momento col comando **File | Carica progetto**.

La finestra **Salva progetto** contiene tre finestre addizionali nel pannello **Descrizione progetto** posto nella parte bassa della finestra visualizzata. Le finestre mostrano le informazioni contenute nel fle di progetto al momento selezionato dalla finestra **Salva progetto** e che stanno per essere salvate su disco.

Le informazioni contenute in un progetto sono:
- nome e produttore del primo dispositivo selezionato nel progetto
- data e ora di creazione del progetto
- versione del programma con cui il progetto è stato creato
- descrizione testuale del progetto redatta dall'utente (si tratta di testo a piacere, di solito il nome degli autori ed alcune note)

La prima finestra (più in alto) contiene informazioni sul progetto al momento selezionato dalla finestra **Salva progetto**.

La seconda finestra (a metà) mostra informazioni sull'attuale confugrazione del programma compresi i dispositivi attualmente selezionati, lo stato di attività del programmatore, la data, l'ora e la versione del programma. Questi dati vengono inseriti nella descrizione del progetto.

La terza finestra (in basso) è modificabile dall'utente e contiene una descrizione del progetto (testo a piacere), di solito il nome degli autori ed alcune note.

File | Ricarica file

Questa opzione permette di ricaricare un file usato di recente.

Un file, dopo essere stato aperto, viene aggiunto alla lista chiamata **Ricarica file**. I nomi dei files sono mostrati nell'ordine con cui sono stati aperti, ovvero in cima si trovano quelli usati di recente.

Per ricaricare un file:

1. Dal menu File, scegliere **Ricarica file**.
2. Viene mostrata la lista dei files usati di recente, clickare sul nome del file desiderato.

Nota: *Quando un file viene ricaricato, si assume che sia del formato con cui è stato salvato l'ultima volta.*

File | Ricarica progetto

Questa opzione permette di ricaricare un progetto usato di recente.

Un progetto, dopo essere stato aperto, viene aggiunto alla lista chiamata **Ricarica progetto**. I nomi dei progetti sono mostrati nell'ordine con cui sono stati aperti, ovvero in cima si trovano quelli usati di recente.

Per ricaricare un file:

1. Dal menu File, scegliere **Ricarica progetto**.
2. Viene mostrata la lista dei progetti usati di recente, clickare sul nome del file desiderato.

File | Opzioni progetto...

Questa opzione viene usate per esaminare/modificare le informazioni del progetto attualmente caricato. Queste comprendono una descrizione essenziale del progetto ed i seguenti dati:

- nome dispositivo e nome produttore
- data creazione progetto
- versione del programma sotto cui il progetto è stato creato
- definizione del progetto fornita dall'utente (testo a piacere), ad esempio i nomi degli autori ed ulteriori informazioni che descrivono il progetto

L'utente può modificare a piacimento il solo testo di descrizione e definizione del progetto. Le altre informazioni sono aggiornate automaticamente dal programma.

File | Carica e table...

Questo comanda carica i dati da un file binario su disco e li salva nella parte della memoria riservata alla tavola di criptazione (di sicurezza).

File | Salva e table...

Questo comanda salva i dati in un file binario su disco prendendoli dalla parte della memoria riservata alla tavola di criptazione (di sicurezza).

File | Esci senza salvare...

Questo comando dealloca lo heap, cancella il buffer su disco (se esiste) e restituisce le risorse al sistema operativo.

File | Esci e salva...

Questo comando dealloca lo heap, cancella il buffer su disco (se esiste), salva nel file di configurazione le impostazione relative agli ultimi 10 dispositivi usati e restituisce le risorse al sistema operativo.

BUFFER

Permette di manipolare il buffer compiendo operazioni su blocchi di memoria, riempiendone una parte con una stringa, cancellandolo, calcolando varie checksum e naturalmente visualizzando e modificando ogni singolo valore (inoltre si può cercare una stringa, stampare il buffer, etc.).

Buffer | Mostra/Cambia... <F4>

Visualizza (View) o modifica (Edit) ogni singolo dato nel buffer (visualizza solo in DUMP mode). Usate i tasti cursore per selezionare il dato da modificare, un dato modificato viene mostrato con colore diverso.
Le seguenti funzioni (attivate da tasti e dai menu) sono richiamabili dagli appositi pulsante presenti nella finestra di Mostra/Cambia Buffer.

<F1>	mostra la schermata di aiuto
<F2>	riempie il buffer dall'indirizzo iniziale a quello finale con la stringa (ASCII o esadecimale) specificata.
<Ctrl+F2>	cancella il buffer riempiendolo col valore specificato
<Ctrl+Shift+F2>	riempie il buffer con valori casuali
<F3>	copia una sequenza di bytes da un indirizzo iniziale ad uno finale a partire da un terzo indirizzo specificati dall'utente. Il terzo indirizzo può essere incluso tra i primi due.
<F4>	sposta una sequenza di bytes da un indirizzo iniziale ad uno finale a partire da un terzo indirizzo specificati dall'utente. Il terzo indirizzo può essere incluso tra i primi due.
<F5>	scambia i bytes alti con i byte bassi nel campo di indirizzi specificato. L'indirizzo iniziale deve essere pari ed il numero di bytes compresi deve essere pari. Se questa condizione non viene rispettata, il programma prende come indirizzo iniziale il primo pari precedente a quello specificato e come indirizzo finale il primo pari successivo a quello specificato.
<F6>	stampa il contenuto del buffer
<F7>	stringa da trovare (massimo 16 caratteri ASCII)
<F8>	stringa da trovare e da sostituire (massimo 16 caratteri ASCII)
<F9>	vai ad un altro indirizzo
<F10>	communta tra visualizza e modifica
<F11>	commuta tra 8 e 16 bit. Può essere fatto anche con il mouse clickando i pulsanti a destra di Mostra/Cambia. Tali pulsanti indicano anche il modo di visualizzazione attuale (8 o 16 bit).
<F12>	calcola i vari checksum disponibili
<Tasti freccia>	spostano il cursore
<Home/End>	salta all'inizio / alla fine della linea corrente
<PgUp/PgDn>	salta alla pagina precedente / successiva
<Ctrl+PgUp/PgDn>	salta all'inizio / alla fine delal pagina corrente
<Ctrl+Home/End>	salta all'inizio / alla fine del contenuto del dispositivo attuale
<Backspace>	sposta il cursore indietro di una posizione (a sinistra)

Nota: *i caratteri 20H - FFH (ASCII) ed i numeri 0..9, A..F (esadecimali) modidicano immediatamente il contenuto del buffer.*
Attenzione: *Non si possono gestire in ASCII i dati con dispositivi a 16 bit.*

Stampa buffer
Invia la parte selezionata del buffer su stampante o su un file. Il programma utilizza un editor di testi esterno nel quale viene visualizzata la parte selezionata del buffer e dal quale questa può essere salvata o stampata. Di default viene usato il semplice editor di testi **Notepad.exe**, il quale viene fornito con qualunque versione di Windows.
In questa finestra le opzioni sono:
- Definisci l'indirizzo iniziale del blocco selezionato nel buffer.
- Definisci l'indirizzo finale del blocco selezionato nel buffer.

Trova testo
Inserite la stringa di ricerca da trovare **(Testo da trovare)** e premete il pulsante **Trova** per iniziare la ricerca o scegliete il pulsante **Annulla** per lasciare perdere.
Direzione specifica in quale direzione deve avvenire la ricerca: **Avanti** a partire dalla posizione corrente del cursore è il default. **Indietro** oltrepassal'inizio del buffer. In visualizzazione cerca in tutto il buffer.
Origine specifica da dove deve iniziare la ricerca.

Sostituisci testo
Inserite la stringa da cercare nella casella **Testo da trovare** e la stringa con cui verrà sostituita nella casella **Sostituire con**.
In **Opzioni** potete scegliere se la sostituzione avverrà automaticamente o se volete essere avvertiti prima cha avvenga per poterla annullare.
Origine indica dove deve iniziare la ricerca.
Direzione specifica in quale direzione deve avvenire la ricerca: **Avanti** a partire dalla posizione corrente del cursore è il default. **Indietro** oltrepassal'inizio del buffer. In visualizzazione cerca in tutto il buffer.
Premete **<Esc>** o clickate sul pulsant **Annulla** per chiudere la finestra.
Premendo il pulsante **Sostituisci** la finestra viene chiusa ed appare un questionario che contiene le seguenti scelte:

Si	sostituisci l'elemento trovato e cerca il prossimo
No	non sostituire l'elemento trovato e cerca il prossimo
Sostituisci tutto	sostituisci tutto
Ferma ricerca	cancella questa operazione

Buffer | Riempi blocco...

Riempie il blocco selezionato del buffer con il valore specificato. Indicare l'indirizzo iniziale e finale del blocco ed il valore di riempimento.

Buffer | Copia blocco...

Copia il blocco selezionato del buffer a partire dall'indirizzo specificato. Indicare l'indirizzo iniziale e finale del blocco ed l'indirizzo di destinazione, anche compreso tra i primi due.

Buffer | Sposta blocco...

Sposta il blocco selezionato del buffer a partire dall'indirizzo specificato. Indicare l'indirizzo iniziale e finale del blocco ed l'indirizzo di destinazione, anche compreso tra i primi due. Il blocco iniziale verrà riempito con il carattere di vuoto.

Buffer | Cambia dati nel blocco...

Scambia i bytes alti con i byte bassi nel campo di indirizzi specificato. L'indirizzo iniziale deve essere pari ed il numero di bytes compresi deve essere pari. Se questa condizione non viene rispettata, il programma prende come indirizzo iniziale il primo pari precedente a quello specificato e come indirizzo finale il primo pari successivo a quello specificato.

Buffer | Cancellazione... <Ctrl+F2>

Riempie il buffer con il carattere di blank specificato dall'utente.
La combinazione **<Ctrl+F2>** farà apparire questo menu in ogni momento.

Buffer | Riempimento casuale... <Shift+Ctrl+F2>

Riempie il contenuto del buffer con valori casuali.
La combinazione **<Shift+Ctrl+F2>** farà apparire questo menu in ogni momento.

Buffer | Duplicazione contenuto del buffer...

Duplica il contenuto del buffer da una EPROM in una EPROM destinazione più ampia. Questa procedura può, ad esempio, copiare in una 27C512 il contentuo di una 27C256 nella posizione desiderata.

Nota: *L'indirizzo iniziale del buffer è sempre 00000h.*

Buffer | Checksum...

Calcola il checksum nel blocco del buffer selezionato. Il checksums viene calcolato con i seguenti algoritmi :

Byte	somma di bytes ad una “word”. Il flag CY viene ignorato
Word	somma di word ad una “word”. Il flag CY viene ignorato
Byte (CY)	somma di bytes ad una “word”. Il flag CY viene aggiunto al risultato
Word (CY)	somma di word ad una “word”. Il flag CY viene aggiunto al risultato
CRC-CCITT	somma bytes a “word” con VAL=PRECEDENTE+(x^16 + x^12 + x^5 +1)
CRC-XModem	somma bytes a “word” con VAL=PRECEDENTE+(x^16 + x^15 + x^2 +1)

La colonna **Neg.** è il complemento a uno della somma cosicché Somma + Neg. = FFFFH.
La colonna **Suppl.** è il complemento a due della somma cosicché Somma + Suppl. = 0 (+ carry).

La finestra di checksum contiene i seguenti oggetti:

Da indirizzo: Indirizzo del primo byte del blocco selezionato sul quale viene calcolato il checksum.

A indirizzo: Indirizzo dell'ultimo byte blocco del selezionato sul quale viene calcolato il checksum.

Inserire checksum: Permette di selezionare il tipo di checksum verrà scritta nel buffer all'esecuzione di **Calcolare e inserire**.

Inserire all'indirizzo: Permette di impostare la dimensione della checksum calcolata che verrà scritta nel buffer all'esecuzione di **Calculare e inserire**. L'indirizzo non si può specificare come un intervallo tipo **Da indirizzo** fino ad **A indirizzo,** ma come indirizzo di Byte.

Tipo: Determina la dimensione della checksum che verrà scritta nel buffer. Tale dimensione può essere 8 (byte) o 16 (word) bits. Se viene selezionato 16 l'intera checksum verrà scritta nel buffer, altrimenti verrà scritto solo il byte basso.
Nota: *Se viene selezionata 16 come dimensione verrà scritto nel buffer il byte basso della checksum all'indirizzo specificato mentre il byte alto verrà scritto all'indirizzo successivo.*

Calcola: Clickando su questo pulsante si effettua il calcolo della checksum per il blocco selezionato nel buffer. Non viene scritto nulla dentro il buffer.

Calcolare e inserire: Clickando su questo pulsante si effettua il calcolo della checksum per il blocco selezionato nel buffer. Il valore viene scritto nulla dentro il buffer all'indirizzo specificato da **Inserire all'indirizzo**.

Chiudi: Clickando su questo pulsante si chiude la finestra Checksum.

COMPONENTE

Questo menu contiene le funzioni che servono per selezionare e lavorare con un dispositivo programmabile, ovvero selezione, programmazione, verifica e cancellazione.

Componente | Scelta/default... <F5>

Questa finestra permette di selezionare il dispositivo desiderato da una lista di dispositivi di default. La lista è ciclica e contiente gli ultimi 10 dispositivi selezionati comprese le opzioni di configurazione. La lista può essere salvata col comando **File | Esci e salva**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete ma combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzaazione, l'algoritmo di programmazioneed una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Usate il tasto **<Del>** per cancellare il dispositivo corrente dalla lista. Non è possibile svuotare completamente la lista. Sull'ultimo dispositivo il tasto **<Del>** viene ignorato.

Componente | Scelta componente ... <Alt+F5>

Questa finestra permette di selezionare il dispositivo desiderato dalla lista di quelli supportati dal programmatore attivo. I criteri sono **Tutti**, **Solo tipo selezionato** o **Solo produttore scelto**.

Il dispositivo selezionato viene salvato automaticamente nella lista dei 10 dispositivi di default raggiungibile mediante il comando **Componente | Scelta default...**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete ma combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzaazione, l'algoritmo di programmazioneed una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Scelta componente | Tutti

Questa finestra permette di selezionare il dispositivo desiderato dalla lista completa dei dispositivi supportati dal programmatore attivo. I nomi vengono mostrati in una casella a scorrimento.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Spazio>** come separatore) e premete **<Enter>** o clickate il pulsante **OK**.

Premete **<Esc>** o clickate il pulsante **Annulla** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete ma combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzaazione, l'algoritmo di programmazioneed una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Scelta componente | Solo tipo selezionato

Questa finestra permette di selezionare il tipo di dispositivo desiderato (ad esempio EPROM) ed il sottotipo (esempio 64Kx8 (27512)), tramite il mouse o i tasti cursore. Verrà visualizzata una lista di produttori e dispositivi.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Spazio>** come separatore) e premete **<Enter>** o clickate il pulsante **OK**.

Premete **<Esc>** o clickate il pulsante **Annulla** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete ma combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzaazione, l'algoritmo di programmazioneed una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Scelta componente | Solo produttore scelto

Questa finestra permette di selezionare il dispositivo desiderato di un particolare produttore, tramite il mouse o i tasti cursore. Verrà visualizzata una lista di produttori e dispositivi.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Spazio>** come separatore) e premete **<Enter>** o clickate il pulsante **OK**.

Premete <**Esc**> o clickate il pulsante **Annulla** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete ma combinazione di tasti <**Ctrl+F1**>. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazioneed una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Componente | Scelta EPROM /FLASH per ID... <Ctrl+F5>

Si usa questo comando per selezionare automaticamente una EPROM o una Flash leggendo l'ID del dispositivo. Il programmatore può identificare il dispositivo leggendo l'ID del produttore e del dispositivo stesso “marchiato” dentro il chip. Questa possibilità esiste solo per EPROM o Flash che la supportano. Se il dispositivo non supporta questo metodo di riconoscimento, un messaggio che indica un dispositivo sconosciuto o non supportato verrà mostrato.

Se più di un dispositivo dello stesso costruttore condividono lo stesso ID verrà mostrata la lista contutti i loro nomi. Quello desiderato può essere selezionato posizionandosi sul suo nome e premendo <**Enter**> (o clickando il pulsante **OK**). Premete <**Esc**> o clickate il pulsante **Annulla** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

ATTENZIONE: *Il programma di conrtollo supporta solo memorie con massimo 40 piedini. Tutti i programmatori* grifo® *determinano automaticamente il numero dei piedini. Con altri programmatori dovreste inserire manualmente il numero dei piedini.*

Il programmatore applica una tensione elevata agli appositi contatti dello zoccolo. Questo è necessario per attivare il sistema di lettura degli ID. Non inserite nello zoccolo un dispositivo diverso da una EPROM o da una Flash, potrebbe essere danneggiato dalla tensione elevata.

Vienesconsigliato di applicare questo comando ad EPROM di piccole dimensioni (tipo 2764 e 27128) perchè quasi tutte non supportano la lettura dell'ID.

Componente | Opzioni componente

Questo menu imposta parametri di programmazione, serializzazione e gestione dei file associati. Il menu **Componente | Opzioni componente | Opzioni operative** contiene sempre elementi specifici del dispositivo selezionato, per avere ulteriori informazioni si veda la finestra delle informazioni specifiche del dispositivo (ottenibile premendo <**Ctrl+F1**>).

Componente | Opzioni componente | Opzioni operative... <Alt+O>

Le impostazioni di questo comando sono usate per controllare il processo di programmazione. Questo è un ambiente flessibile che contiene impostazioni associate al dispositivo selezionato, quelle non supportate dal programmatore attivo sono disabilitate. Le impostazioni vengono salvate su disco eseguendo il comando **File | Esci e salva**.
In generale l'utente decide quali operazioni effettuare, oppure quali aree del componente programmare in base alle sue esigenze d'uso del dispositivo da programmare.
Di seguito vengono elencate e brevemente descritte la lista delle impostazioni, suddivise nei loro gruppi di appartenenza:

- gruppo **Indirizzi**:

Inizio componente	(default 0)
Fine componente	(default dimensione-1)
Inizio buffer	(default 0)
Split	(nessuno)

- gruppo **Controllo inserimento:**

Controllo inserimento	(default abilitato)
Riconoscimento ID	(default abilitato)

- gruppo **Sequenza comandi:**

cancella prima di programmare	(default disabilitato)
verifica se è vuoto	(default disabilitato)
verifica dopo la lettura	(default disabilitato)
verifica dopo la programmazione	(una volta)
opzioni verifica	(VCC nominale)

Componente | Opzioni componente | Serializzazione...

La serializzazione è una particolare modalità di programmazione. Quando è attivata, un particolare codice numerico viene automaticamente scritto ad un indirizzo specificato del buffer prima di ogni programmazione. Quando molti dispositivr vengono programmati singolarmente, il numero di serie viene cambiato automaticamente prima della programmazione, così ogni dispositivo ha un numero di serie univoco.
Ci sono tre tipi di serializzazione:

- Disabilitata
- Incrementale
- Da file

Quando viene selezionato un nuovo dispositivo la serializzazione viene automaticamente disabilitata. Gli attuali parametri di serializzazione per il dispositivo corrente vengono salvati nella configurazione generale con il comando **File | Esci e salva**.
Quando è abilitato il modo incrementale le seguenti configurazioni vengono salvate su file: indirizzo, dimensione, numero seriale, passo dell'incremento e le impostazioni di ASCII / BIN, DEC / HEX, LS / MS Byte .
Quando è abilitato il prelievo dei numeri da file le seguenti configurazioni vengono salvate: nome del file dei numeri ed etichetta corrente, quest'ultima indica la linea del file dalla quale verrà preso il prossimo numero.

Componente | Opzioni componente | Serializzazione | Modo incrementale

La modalità incrementale permette di assegnare numeri di serie individuali ad ogni dispositivo che viene programmato. Un valore iniziale viene incrementato ad ogni dispositivo programmato correttamente e scritto nel formato selezionato all'indirizzo del buffer desiderato.
L'utente può modificare le seguenti opzioni in modalità incrementale:

Dimensione S/N

definisce la lunghezza in bytes del numero di serie che verrà scritto nel buffer. Per serializzazioni binarie i valori validi sono da 1 a 4, mentre per le serializzazioni in ASCII i valori validi sono da 1 ad 8.

Indirizzo

Specifica l'indirizzo nel buffer dove il numero di serie viene scritto. Si noti che tale valore deve essere compreso nel campo di indirizzi tra quello iniziale e quello finale e deve essere

specificato correttamente, ovvero l'intero numero di serie deve ricadere entro il campo di indirizzi validi.

Valore iniziale

Specifica il valore iniziale dei numeri di serie. Generalmente il valore massimo per serializzazioni binarie con 4 bytes è $1FFFFFFF.

Quando il valore del numero di serie supera il massimo vengono azzerati i tre bit più significativi del numero di serie corrente. Dopo tale azione il numero si trova ad essere sempre compreso tra 0 e $1FFFFFFF (gestione degli overflow secondo lo stile del basic).

Passo

Specifica il passo di incremento del numero di serie.

Modo S/N

Definisce la forma con la quale il numero di serie viene scritto:

- ASCII: Significa che il numero verrà scritto come stringa ASCII. Per esempio il numero di serie $0528CD in ASCII viene scritto nel buffer come la sequenza di bytes 30h 35h 32h 38h 43h 44h (‘0’ ‘5’ ‘2’ ‘8’ ‘C’ ‘D’), ovvero 6 bytes.
- Binario: Significa che il numero di serie viene scritto direttamete nel buffer. Se il numero è composto da più di un byte, ci sono due possibili ordini di scrittura. L'ordine di scrittura può essere cambiato con il comando “Save to buffer” item.

Stile

Definisce la base per i numeri di serie, che può essere:

- Decimale: i numeri vengono inseriti e mostrati usando i caratteri da ‘0’ a ‘9’.
- Esadecimale: i numeri vengono inseriti e mostrati usando anche i caratteri da ‘A’ ad ‘F’.

Vi è il caso speciali dei numeri decimali codificati in binario, o BCD. Ciò significa che ogni cifra decimale viene memorizzato in una cifra esadecimale, ovvero ogni nibble vale da 0 a 9, non sono ammessi i valori da A ad F in un nibble BCD.

La base va scelta con l'opzione “Style” prima di inserire il numero di serie iniziale ed il passo.

Salva nel buffer

Specifica l'ordine di scrittura dei bytes del numero di serie. Questa opzione viene usata quando il numero di serie è di tipo binario, non si applica ai numeri ASCII.

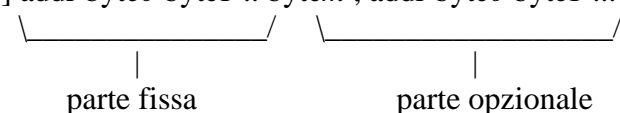
Vi sono due opzioni:

- LSByte first (usato dai processori Intel) posizionerà il byte meno significativo all'indirizzo più basso del buffer.
- MSByte first (usato dai processori Motorola) posizionerà il byte più significativo all'indirizzo più basso del buffer.

Componente | Opzioni componente | Serializzazione | Modo da file

In modalità di serializzazione dal file i numeri vengono letti da un file specificato dell'utente e scritti nell'indirizzo del buffer indicato dal file.

La modalità prevede due impostazioni:

Nome file

Specifica il nome del file da cui vengono letti gli indirizzi ed i relativi numeri di serie. Il file per la serializzazione in questa modalità deve avere un formato specifico, che viene descritto nella sezione più sotto.

Etichetta inizio

Definisce l'etichetta iniziale all'interno del file di numeri. La lettura dei valori dei numeri di serie inizia da questa etichetta.

Il file per la serializzazione da file comprende indirizzi e numeri di serie, questi ultimi specificati come gruppi di bytes. Il file è di tipo testuale, la sua struttura è:

legenda:

parte fissa

La parte fissa definisce l'indirizzo nel buffer ed i bytes che vanno scritti a partire da tale indirizzo. Tali dati vanno sempre scritti dopo l'etichetta.

parte opzionale

La parte opzionale definisce un secondo gruppo di indirizzo e bytes da scrivere nel buffer. Una parte opzionale può essere definita dopo la parte fissa dei dati.

etichette (label*1*, label*n* - labels)

Le etichette identificano ogni linea del file in ingresso. Vengono usate per idnirizzare individualmente ogni riga del file, per cui ogni etichetta dovrebbe essere univoca. Lo scopo è indirizzare la linea da cui inizia il conteggio dei numeri di serie

addr -

Definisce l'indirizzo nel buffer da cui inizare a scrivere il numero di serie.

byte*0*..byte*n*, byte*0*..byte*m*, byte*0*..byte*k* -

I gruppi di bytes byte*0*..byte*n*, byte*0*..byte*m* e byte*0*..byte*k* sono i dati che verranno scritti nel buffer. Al massimo ci possono essere 64 bytes in un campo dati seguente ad un indirizzo.

I dati vengono scritti nel buffer dall'indirizzo addr all'indirizzo addr+n:

byte*0* in addr
byte*1* in addr + *1*
byte*2* in addr + *2*
....
byte*n* in addr + *n*

La parte opzionale viene separata dal carattere “ , “ (virgola) ed ha la stessa struttura della parte precedente, ad esempio un indirizzo ed un gruppo di bytes.

Caratteri ad uso speciale per etichette:

[ ] - le etichette devono essere racchiuse da parentesi quadre
‘,’ – carattere separatore la parte fissa e quella opzionale
‘;’ - il carattere di punto e virgola indica l'inizio di un commento. Tutti i caratteri dal punto e virgola alla fine della linea vengono ignorati. Un commento può anche essere a fine linea.

Note:

- *Un nome di etichetta può contentere tutti i caratteri tranne ‘[‘ e ‘]’. I nomi di etichetta non tengono conto della capitalizzazioe, ovvero ‘a’ è uguale ad ‘A’, ‘b’ è uguale a ‘B’ ecc..*
- *Tutti gli indirizzi ed i valori da scrivere devono essere in esadecimale.*
- *Gli indirizzi possono essere composta da 1 fino a 4 bytes.*
- *I gruppi di bytes di dati possono essere composti da 1 a 64 elementi. Quando ci sono due gruppi su una stessa linea, il numero totale di elementi deve essere al massimo 80.*
- *Fate attenzione ad impostare l'indirizzo corretto. Questo deve essere definito nel campo tra gli indirizzi iniziale e finale del dispositivo che viene programmato. Se l'indirizzo fosse fuori da questo campo apparirebbe una finestra e si la serializzazione verrebbe disabilitata.*

Esempio:
[nav*1*] A7890 78 89 56 02 AB CD ; commento1
[nav*2*] A7890 02 02 04 06 08 0A
[nav*3*] A7890 08 09 0A 0B A0 C0 ; commento 2
[nav*4*] A7890 68 87 50 02 0B 8D
[nav*5*] A7890 A8 88 59 02 AB 7D

;la linea successiva contiene anche la parte opzionale
[nav*6*] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16
; ultima linea - fine del file

Nel file di esempio vengono definiti sei valori di numeri di serie etichettati con “nav*1*”, “nav*2*”, ...“nav*6*” . ogni numero viene scritto all'indirizzo $A7890 ed è composto da 6 bytes. La linea etichettata con “nav*6*” ha anche la parte opzionale, che viene scritta nel buffer dall'indirizzo $FFFF6 ed è di 10 bytes, quindi il suo ultimo byte viene scritto all'indirizzo $FFFFF.

Componente | Opzioni componente | Statistiche...

Le statistiche forniscono informazioni circa l'attuale conteggio delle operazioni su un particolare dispositivo. Se su ogni dispositivi viene effettuata una sola operazione allora il conteggio delle operazioni effetuate coincide con il conteggio dei dispositivi programmati.

Un'altra funzione statistica è il conto alla rovescia. Questa permette di controllare sia il numero di operazioni che il numero di dispositivi sui quali vengono effettuate. Dopo ogni operazione andata a buon fine su un certo dispositivo il contatore viene decrementato. L'utente deve stabilire il valore iniziale del conto alla rovescia, quando questo raggiunge lo zero si intende che è stato raggiunto il limite del lotto e viene mostrato un messaggio di conteggio completato.

La finestra **Statistiche** contiene le seguenti opzioni:

Le caselle spuntabili **Programma**, **Verifica**, **Verifica cancellazioni**, **Cancella** e **Lettura** decidono le operazioni che, se andate buon fine, incrementano il conteggio.

Le caselle spuntabili **Count Down** gestisce l'attività di conto alla rovescia (lo abilita o disabilita). La casella di testo accanto alla casella spuntabile definisce il valore iniziale del conteggio.

La finestra **Stato attuale** si può aprire anche premendo il tasto destro del mouse sul pannello Statistiche e clickando sulla voce visualizzata.

I valori delle statistiche vengono mostrati nella finestra principale, nella casella Statistiche. Quest'ultimo contiene in ordine i contatori di **Successi**, **Fallimenti**, **Totali** e l'attuale contatore a rovescia, se abilitato.

Le operazioni completate con successo sono una qualunque tra le seguenti:
- programmazijone
- verifica di programmazione
- verifica se è vuoto
- cancellazione
- lettura

Se una opeazione viene completata con errori non la si considera completata con successo.
Quando viene selezionato un nuovo dispositivo, tutti i valori delle statistiche vengono azzerati e **Count down** viene disabilitato.
Il pulsante **Reset** nel pannello e casella **Statistiche** azzera i valori delle statistiche.
Il pulsante **Ricarica Count down** nella casella **Statistiche** ricarica il valore iniziale di **Count down**.

Componente | Opzioni componente | File associato...

Associa un file al dispositivo corrente. Il file viene caricato automaticamente nel buffer non appena il dispositivo corrente viene di nuovo selezionato dalla lista dei dispositivi o se era già selezionato all'avvio del programma.
Il nome (completo di percorso) viene modificato nell'apposita casella di testo. Il programma controlla automaticamente la presenza del file, il caricamento si può abilitare o disabilitare.
Entrambi questi settaggi (nome del file ed abilitazione del caricamento) si possono salvare su disco con il comando **File | Esci e salva**.

Componente | Verifica cancellazione <F6>

Verifica se un dispositivo è vuoto, completamente o in parte, a seconda del dispositivo. Il risultato dell'operazione viene comunicato da una finestra di avviso.

La finestra **Componente | Opzioni componente | Opzioni operative** permette di modificare l'area standard del dispositivo che viene verificata da questo comando.

Componente | Lettura <F7>

Legge il contenuto di un dispositivo, completamente o in parte, a seconda del dispositivo. Il termine dell'operazione viene comunicato da una finestra di avviso.

La finestra **Componente | Opzioni componente | Opzioni operative** permette di modificare l'area del dispositivo che viene letta da questo comando. Abilitare la verifica dopo la lettura (comando **Verifica dopo lettura**) in questo menu aumenta l'affidabilità della lettura stessa.

Componente | Verifica <F8>

Confronta il contenuto di un dispositivo con il buffer, completamente o in parte, a seconda del dispositivo. Il risultato dell'operazione viene comunicato da una finestra di avviso.

La finestra **Componente | Opzioni componente | Opzioni operative** permette di modificare l'area sulla quale lavora questo comando.

Il comando **Opzioni | Opzioni generali | Errori** permette di scrivere le eventuali differenze sullo schermo o sul file VERIFY.ERR. Sullo schermo vengono mostrate al massimo le prime 45 differenze complete degli indirizzi.

Componente | Programma <F9>

Programma il contenuto di un dispositivo, completamente o in parte, a seconda del dispositivo. L'esito dell'operazione viene comunicato da una finestra di avviso.

La finestra **Componente | Opzioni componente | Opzioni operative** permette di specificare un'altra area di lavoro e di impostare ulteriori elementi della sequenza di operazioni di programmazione.

Componente | Cancellazione <F10>

Cancella il contenuto di un dispositivo. L'esito dell'operazione viene comunicato da una finestra di avviso..

Componente | Test

Esegue il test dei dispositivi indicati nella lista di quelli che possono ammettere un test (ad esempio RAM statiche) col il programmatore attuale.

Componente | IC test

Attiva la sezione di test per circuti logici che si trova separate sul mini-CD di distribuzione per compatibilità. Prima si deve selezionare la libreria appropriata, poi il dispositivo desiderato, infine una modalità del test tra continua e a passo singolo. La sequenza di controllo e il risultato del test vengono mostrati in finestra.

Componente | JAM/VME/... Player

Esistono strumenti propietari realizzati dai maggiori produttori di circuiti integrati al mondo che permettono di svolgere particolari attività di debugging. Alcuni programmatori sono in grado di integrarsi con tali strutture propietarie e di gestirne le funzioni tramite questo menu.
Trattandosi di strumenti speciali, la loro descrizioni esula dagli scopi di questo manuale.

Componente | Informazioni componente... <Ctrl+F1>

Fornisce informazioni aggiuntive sul dispositivo attuale (dimensione, organizzazione, algoritmo di programmazione e una lista dei programmatori che lo supportano, corredata degli eventuali moduli adattatori). Si trovano inoltre informazioni sul package ed eventuali ulteriori informazioni specifiche.

PROGRAMMATORE

Questo menu contiene le funzioni che servono per interagire con il programmatore, ovvero gestire la comunicazione, la diagnostica interna e le funzioni di lettura e programmazione.

Programmatore | Trova programmatore... <Ctrl+F>

Cerca un altro programmatore o ripristina la comunicazione con il programmatore già collegato. Offre le seguenti opzioni:

Programmatore - imposta uno tra i programmatori supportati dal programma di controllo. Se viene selezionata la voce “Search all” verranno cercati tutti.
Stabilire communicazione - stabilisce la comunicazione con un nuovo programmatore o ripristina la comunicazione con quello già collegato.
Velocità - Modificare la velocità di comunicazione è importante con i PC che hanno porte parallele “lente” o che non hanno sufficiente potenza per pilotare il cavo di comunicazione PC<->programmatore (laptop, notebook, ...). Usatie questa opzione per tentare di risolvere eventuali problemi di comunicazione tra PC e programmatore (esempio, il programma di controllo indica che il programmatore è assente la comunicazione col programmatore non è affidablie, ecc.).
Se è stata abilitata la comunicazione automatica allora il programma di controllo imposta la massima velocità possibile.
Porta - sceglie la porta dove verrà cercato il programmatore. Se la voce “Tutte le porte” viene selezionata allora il programmatore verrà cercato su tutte le porte presenti agli indirizzi standard.
Indirizzo speciale - Imposta l'indirizzo per la porta parallela special , se selezionata.
Premendo **<Enter>** o il pulsante **OK** parte la scansione basata sui precedenti parametri. La stessa scansione avviene all'avvio del programma di controllo. Il comando cancella la lista dei dispositivi di default ma non il dispositivo corrente se il nuovo programmatore lo supporta.

Queste impostazioni vengono salvate col comando **Opzioni | Salva opzioni**

Programmatore | Ritrova programmatore <Shift+Ctrl+F>

Cerca un programmatore o ripristina la comunicazione con il programmatore già collegato e che è stato specificato con il menù precedente. Offre le opzioni descritte nel precedente comando.

Programmatore | Handler...

La finestra **Handler** permette di definire un dispositivo esterno che viene abilitato ad effettuare un controllo speciale ed a controllare delle attività effettuate sui dispositivi programmabili. Quando non viene selezionato alcun Handler il programma di controllo funziona come da default, ovvero l'utente controlla direttamente le attività, altrimenti il programma di controllo si trova in modalità speciale, quindi le operazioni sui dispositivi sono controllate in cooperazione con un dispositivo esterno.
La finestra **Handler** contiene le seguenti opzioni:

Seleziona Handler selezione l'Handler desiderato.
Cerca alla porta seleziona una porta seriale (COM) dove verrà cercato l'Handler.

Premendo **<Enter>** o il pulsante **OK** si attiva la ricerca. Se l'Handler selezionato è di tipo **Nessuno**, non verrà effettuatta alcuna ricerca. Le attuali impostazioni sull'Handler vengono salvate su disco mediante il comando **Opzioni | Salva opzioni** o quando il programma di controllo viene chiuso.

L'Handler non è disponibile per le vendita.

Programmatore | Opzioni modulo...

Viene usato con un modulo multi-zoccolo per definire lo zoccolo MASTER e il tipo di attività per ogni zoccolo. Il gruppo di opzoni **Zoccolo MASTER** permette di stabilire quale zoccolo viene usato per le operazioni di lettura. Le caselle sputabili **Abilita/Disabilita zoccolo** abilitano o disabilitano individualmente gli zoccoli. Lo zoccolo disabilitato non effettua alcuna operazione.

Programmatore | YES! automatico...

Imposta la modalità **YES! automatico** con cui basta inserire correttamente un dispositivo nello zoccolo e l'ultima operazione effettuata verrà ripetuta. Automaticamente, il programma rileva l'inserzione di un nuovo dispositivo nello zoccolo ed esegue l'ultima operazione effettuate senza dover premere alcun tasto o pulsante. Viene visualizzata sullo schermo l'avvenuta inserzione di un nuovo dispositivo. La modalità può essere annullata premendo il tasto **<Esc>** durante l'attesa di inserzione di un nuovo dispositivo.

Nota: *Durante l'attesa di un nuovo dispositivo il LED BUSY lampeggia.*

Se viene selezionato un nuovo programmatore con **Opzioni | Trova programmatore**, questa modalità viene disabilitata.

Con **Tempo di risposta** è possibile impostare un intervallo di tempo entro il quale deve essere inserito il dispositivo nuovo. Di default viene preimpostato un intervallo standard. Se viene usato uno zoccolo adattatore di formato, si consiglia di aumentare tale intervallo.

Con **Pins con capacità** si può specificare una lista di pins che vengono connessi attraverso delle capacità (ad esempio se viene usato uno zoccolo convertitore con un condensatore inserito tra Vcc e GND), che potrebbe creare problemi durante l'inserzione di un nuovo dispositivo.
Tale lista deve essere nella forma:
    pinA, pinB, pinC....
Esempio: 4,6,17

La lista viene svuotata se viene eseguito il comando **Componente | Scelta/default** o **Componente | Scelta componente**...

Queste impostazioni vengono salvate sul disco con il comando **Opzioni | Salva opzioni.**

Programmatore | Selftest...

Esegue una procedura di auto diagnostica senza il relativo POD (presente nella confezione). **LO ZOCCOLO ZIF DEVE ESSERE VUOTO PRIMA DI ESEGUIRE QUESTO COMANDO**.
Si consiglia di eseguire anche il comando di auto diagnostica **Programmatore | Selftest plus**.

Programmatore | Selftest plus...

Esegue una procedura di auto diagnostica usando il relativo POD (presente nella confezione). Si consiglia di eseguire questo comando periodicamente, ad esempio una volta ogni sei mesi.

Programmatore | Selftest connettore ISP...

Esegue una procedura di auto-diagnostica per la struttura di programmazione ISP, usando un apposito POD di test. Prima di fornire questo comando si devono eseguire gli altri due comandi di test e poi seguire le indicazioni dell'aiuto in linea.

Programmatore | Test calibrazione...

Esegue una verifica di calibrazione su numerosi componenti interni e sulle tensioni usate.

OPZIONI

Permette di verificare e modificare diverse impostazioni di default ed operative.

Opzioni | Opzioni generali...

Permette di controllare le seguenti opzioni.

Opzioni File

Permette di impostare la maschera per i file, attivare il caricamento automatico del file selezionato e selezionare il tipo del file che verrà caricato.
Maschera di formato dei files imposta un filtro per selezionare i nomi dei file in **File | Salva** e **File | Carica** per tutti i formati. La maschera deve contenere almeno un carattere speciale (*, ?).
Estensione di default per file di progetto imposta l'estensione del file usata di default in **File | Carica progetto** e **File | Salva progetto**.
Il gruppo **Quando il file corrente viene modificato da un altro processo** permette di decidere la politica di ricaricamento del file attualmente selezionato. Ci sono tre opzioni:

1. Chiede prima di ricaricare
2. Ricarica sempre
3. Ignora i cambiamenti del file

Formato del file permette di decidere la politica di riconoscimento dei file che vengono caricati. Quando viene messo in automatico, il programma analizza il formato cercando ogni tipo di file supportato. Se il file corrisponde allo schema di validazioni di un formato supportato, verrà caricato nel buffer usando quel formato.
Se messo in manuale l'utente può indicare esplicitamente il formato del file che sta per essere caricato dalla lista dei formati supportati. In caso il formato indicato non corrisponda a quello effettivo del file, ques'ultimo potrà essere caricato parzialmente o non correttamente.

Hex file options

Contiene diverse opzioni per controllare il caricamento dei file in formato HEX.
La prima opzione abilita la cancellazione del buffer (col valore desiderato) che avviene automaticamente prima di caricare il file.
La seconda opzione imposta un offset negativo usato per modificare l'indirizzo di caricamento di ogni file HEX in modo che ricada in un campo di indirizzi validi.
Esempio:
Un file contiene dati in formato Motorola S. Un blocco di dati inizia dall'indirizzo FFFF0H. Il formato è S2 quindi l'indirizzo è un gruppo di 3 bytes. Potete associare un offset negativo di FFFF0H alla lettura di ogni dato. Quindi l'offset verrà sottratto dall'effettivo indirizzo specificato.nel file e i dati verranno caricati a partire dall'indirizzo 0.

Avvertimento: *Il valore di offset viene sottratto all'indirizzo effettivo quindi il risultato portebbe essere un indirizzo minore di 0. Fate molta attenzione ad impostare l'offset negativo.*

Lingua
Permette di selezionare un'altra lingua per l'interfaccia utente ovvero menu, pulsanti, finestre, informazioni e messaggi. Permette anche di avere l'help in un'altra lingua. Per il supporto a questa funzione è necessario che ci sia un file di definizione della lingua.

Suono
Permette all'utente di associare degli effetti sonori al completamento di determinate operazioni come, ad esempio, programmazione, verifica, lettura, etc. Il programma genera un suono anche quando viene mostrato un avvertimento o un errore. L'utente può selezionare un suono dal sistema audio di MS Windows (richiede la corretta installazione di una scheda audio), dallo speaker del PC o disabilitare i suoni.

Altro
Permette di impostare ulteriori parametri del programma.
Si possono disabilitare gli hint (suggerimenti) che appaiono sui pulsanti delle barre posizionandoci sopra la freccia per alcuni secondi infine si può modificare la directory di partenza all'avvio del programma, a scelta tra la directory di installazione e l'ultima directory visitata prima di chiudere il programma.

Salva opzioni
Permette di decidere cosa salva il programma alla chiusura ed offre tre scelte:
- **Non salvare opzioni** e non chiedere se salvare le opzioni.
- **Salva opzioni automaticamente** senza chiedere conferma.
- **Chiede all'utente** se vanno o non vanno salvate le opzioni.

Opzioni | Mostra

Seleziona se mostrare o nascondere vari elementi dell'interfaccia utente come le barre.

Opzioni | Mostra | Barra strumenti principale
Mostra o nasconde la barra dei pulsanti principale.

Opzioni | Mostra | Barre degli strumenti aggiuntive
Mostra o nasconde la barra dei pulsanti secondaria.

Opzioni | Mostra | Opzioni componente prima di operazioni
Mostra o nasconde l'elenco delle opzioni di programmazione per un certo dispositivo prima che la programmazione inizi, che funge da promemoria per l'utente che le può quindi verificare e confermare.

Opzioni | Modo protetto...

La modalità protetta è una speciale modalità del programmatore. In questa situazione sono disabilitate le operazioni di programmazione, quelle che possono modificare il contenuto del buffer o che possono variare la configurazione del dispositivo da programmare. Quindi in questa situazioni l'operatore non può modificare i parametri significativi della programmazione per errore, risulta utile nella programmazione di grandi lotti.

Ci sono due modi per attivare la modalità protetta:

1. con il comando **Opzioni | Modo protetto**. Questo comando richiede la password, l'utente la deve inserire due volte per dare conferma. Dopo l'accettazione della password avviene il passaggio in modalità protetta, la stessa password serve per tornare in modalità normale.
2. aprendo un progetto precedentemente salvato in modalità protetta.

Per tornare in modalità normale usate il comando **Opzioni | Modo normale**. La password che verrà richiesta è la stessa usate per commutare in modalità protetta.
Un altro modo per uscire dalla modalità protetta consiste nel chiudere il programma, in quanto la modalità protetta è attiva solo fino all'uscita dal software. Al prossimo avvio il programma si troverà di nuovo in modalità normale (unica eccezione è il caso di un progetto caricato dalla linea di comando precedentemente salvato in modalità protetta).

Opzioni | Salva opzioni

Questo comando salva tutte le impostazioni al momento abilitate ad essere salvate, persiono se il salvataggio automatico viene disabilitato. Sono salvate le seguenti opzioni: quelle del menu **Opzioni**, gli ultimi dieci dispositivi selezionati, gli ultimi file aperti, posizione e dimensione della finestra principale.

AIUTO

Premendo **<F1>** si accede all'Help. Premendo **<F1>** mentre è selezionato un particolare elemento di un menu, si accede all'help sensibile al contesto. Se PG4UW sta eseguendo una qualche operazione sul programmatore **<F1>** non genera alcuna reazione.

Poichè il sistema di Help viene aggiornato continuamente con il programma di controllo, può contenere informazioni non incluse in questo manuale.
Informazioni dettagliate sui comandi indivuduali si possono trovare nell'Help in linea.

Nota: *Le informazioni sono precise al momento della pubblicazione dello stesso. I nostri prodotti sono aggiornati di continuo. Si prega di consultare il sito www.grifo.it* o *www.grifo.com.*

Aiuto | Contenuti F1

Mostra tutto l'aiuto in linea.

Aiuto | Cerca aiuto per...

Mostra la pagina di aiuto in linea riferita alla finestra, comando, casella, dato, valore, formato, attualmente selezionato dal cursore.

Aiuto | Componenti supportati

Mostra una lista con tutti i dispositivi supportati da almeno un programmatore.
Torna utile quando l'utente vuole trovare un qualunque dispositivo purchè sia supportati da almeno uno di quelli compatibili con il programma di controllo.

Aiuto | Programmatori supportati

Mostra informazioni sui programmatori supportati dal programma di controllo.

Aiuto | Lista componenti (programmatore attuale)

Crea una lista di tutti i dispositivi supportati dal programmatore corrente nel file di testo **?????DEV.txt** e nel file HTML **?????DEV.htm** che vengono creati nella directory dalla quale viene eseguito il programma di controllo. Il segno ????? viene sostituito con una forma abbreviata del nome del programamtore al quale la lista è riferita.

Aiuto | Lista componenti (tutti i programmatori) <Shift+Alt+D>

Crea una lista di tutti i dispositivi supportati da tutti i programmatori grifo® nel file di testo **?????DEV.txt** e nel file HTML **?????DEV.htm** che vengono creati nella directory dalla quale viene eseguito il programma di controllo. Il segno ????? viene sostituito con una forma abbreviata del nome del programamtore al quale la lista è riferita.

Aiuto | Lista componenti (riferimenti incrociati)

Crea una cross reference tra tutti i dispositivi supportati da tutti i programmatori presenti sul mercato e quelli supportati dal programmatore corrente. Il risultato è in formato HTML e consiste dei seguenti file:

- un file principale in HTML **TOP_DEV.htm** che lista i produttori dei dispositivi
- file HTML parziali che listano i dispositivi per ogni produttore

Il file HTML principale viene creato nella stessa directory ove si trova il programma di controllo. I file parziali HTML vengono messi nelle sotto directory **DEV_HTML** che partono dalla directory dove si trova il programma di controllo.

Aiuto | Crea report dei problemi

Crea il report diagnostico con tutte le informazioni interne e le operazioni svolte (ovvero il contenuto della finestra di log) e lo salva in un file di testo sulla scrivania, con il nome **PG4UW_LOG_window_content.txt**. Il report diagnostico è utile quando si verifica un qualche errore nel programma di controllo o nel programmatore che l'utente non è in grado di risolvere e quindi deve contattare il produttore. In questi casi è indispensabile fornire il report diagnostico, in quanto questo può aiutare il produttore a riprodurre e localizzare più rapidamente la causa del problema ed a risolverlo.

Aiuto | Informazioni...

Fa apparire una finestra che mosta le informazioni sul programmatore attuale, lingua scelta, generalità della ditta realizzatrice ed il numero di versione.

CARATTERISTICHE SOFTWARE AGGIUNTIVE

PG4UW è il programma di controllo comune a tutti i programmatori **grifo®**. Quindi potreste trovare alcune voci di menu che non si riferiscono al vostro programmatore.

Si può associare un file particolare ad ogni dispositivo, in modo che venga caricato automaticamente nel buffer se l'opzione **Componente | Opzioni componente | File Associato | Caricamento automatico** è impostata a Si. Il file associato è l'ultimo caricato, a meno che non venga esplicitamente dichiarato nella casella **Componente | Opzioni componente | File Associato | Nome file**.

Una caratteristica denominata **YES! automatico** permette in concreto, di ripetere l'ultima azione eseguita senza bisogno di premere un pulsante o un tasto. Questo è possibile poichè il programmatore determina autonomamente l'inserimento di un nuovo dispositivo nello zoccolo e ripete subito l'ultima operazione. Tutti i parametri di questa caratteristica sono impostabili da menu **Opzioni**. Se usate adattatori di formato che connettono dei condensatori ai pin, inserite tali pin nella apposita lista.

Il pulsante **YES** consente a sua volta di ripetere l'ultima operazione svolta senza dover usare la tastiera del PC, in modo da velocizzare la fase di programmazione di numerosi componenti. Per individuare la posizione di tale pulsante si faccia riferimento alla figura 4.

Alcuni dispositivi speciale (ad esempio i PHILIPS della famiglia Coolrunner) richiedono dei file DAT esterni non forniti di default nel mini CD. Se avete bisogno di programmarli, esaminate la sezione Download dei siti internet **www.grifo.it** o **www.grifo.com**.

Indipendentemente dal sistema operativo Windows usato, siete pregati di non spostare mai alcuna finestra mentre il LED BUSY è acceso: il circuito di protezione del programmatore contro errori di comunicazione o eccessiva lentezza del PC potrebbe attivarsi e mettere il programmatore in modalità di sicurezza. Per la stessa ragione durante le operazioni sui dispositivi é consigliato non sovracaricare il PC in uso con l'esecuzione contemporanea di numerosi altri programmi.

Fate attenzione alla possibilità di Windows di gestire diverse applicazioni in multitasking: per un corretto funzionamento del programma di controllo è indispensabile che la porta parallela a cui è collegato il programmatore sia usata solo dal PG4UW e non da altri applicativi.

Il programma di controllo è in grado di gestire tutti i tipi di porta parallela (pieno supporto IEEE 1284), quindi non c'è bisogno di configurare la porta parallela in modalità particolari.

COME INIZIARE

Questo capitolo contiene tutte le informazioni essenziali per la connessione del programmatore al PC, installare il programma di controllo ed iniziare ad usare il programmatore in brevissimo tempo. **Si prega di leggere completamente questi passi prima di tentare qualunque utilizzo del programmatore**, soprattutto se l'utente non ha mai usato un programmatore. Per soddisfare le esigenze della maggioranza degli utilizzatori il capitolo propone i passi da eseguire per copiare un dispositivo e/o per programmarlo con un file salvato su PC.
Per ottenere informazioni più dettagliate sulle caratteristiche tecniche del programmatore e del software di controllo potete riferirvi ai capitoli precedenti di questo manuale.

a) Installate il software PG4UW come descritto nel paragrafo INSTALLAZIONE PROGRAMMA DI CONTROLLO. Qualora il programma non sia installato dal mini CD, bensì dall'ultima versione scaricata da internet, seguite le indicazioni del paragrafo AGGIORNAMENTO PROGRAMMA DI CONTROLLO.

b) Predisponete l'**UEP 50** come descritto nel paragrafo PREPARAZIONE PROGRAMMATORE.

c) Collegate il programmatore al PC tramite l'interfaccia USB come come descritto nel paragrafo COLLEGAMENTO USB: INSTALLAZIONE DRIVER. In alternativa collegarlo tramite l'interfaccia parallela come descritto nel paragrafo COLLEGAMENTO LPT.

d) Eseguite il programma PG4UW come descritto nel paragrafo ESECUZIONE DEL PROGRAMMA DI CONTROLLO.

e) Scegliete il dispositivo che volete programmare sia tramite il menu **Componente | Scelta componente...**, sia tramite la combinazione di tasti <**Alt+F5**> (premete **<F5>** mentre tenete il tasto **<Alt>** premuto) oppure più semplicemente clicckare l'icona:

f) Caricate i dati con cui programmare il dispositivo selezionato:
- Se volete caricare un file da disco, usate il menu **File | Carica...** o premete il tasto <**F3**> o più semplicemente clicckare l'icona:

- Se volete copiare il contenuto di un dispositivo già programmato, dovete prima leggere il suo contenuto. In questo caso: aprire lo zoccolo ZIF (leva in alto), inserite il dispositivo già programmato facendo attenzione che l'orientamento corrisponda a quello dell'adiacente disegno, chiudere lo zoccolo ZIF (leva in basso) ed effettuare la lettura tramite il menu **Componente | Lettura** o premendo il tasto <**F7**> o più semplicemente clicckare l'icona:

Al termine della lettura, segnalata sia sul programma che dal LED GOOD del programmatore

provvedere ad aprire lo zoccolo ZIF ed ad estrarre il componente letto.
Visto che il programmatore **UEP 50** si rivolge anche alla programmazione di microcontrollori e PLD ricordare che questi possono essere protetti contro la rilettura. Verificate la presenza di questa protezione (contattando il fornitore del dispositivo) e non proseguite se attivata.

g) Inserite il dispositivo da programmare nello zoccolo ZIF sempre rispettando l'allineamento e quindi chiuderlo.

h) Verificate che il componente da programmare sia cancellato e quindi programmabile tramite il menu **Componente | Verifica cancellazione.**, oppure il tasto <**F6**> o più semplicemente clicckare l'icona:

i) Programmate il contenuto attuale del buffer sul dispositivo col menu **Componente | Programma** oppure premendo il tasto <**F9**> o più semplicemente con l'icona:

j) Potete confrontare il contenuto del buffer con i dati appena scritti nel dispositivo da programmare tramite il menu **Device | Verifica** oppure premendo il tasto <**F8**> o più semplicemente con l'icona:

k) Aprire lo zoccolo ZIF (leva in alto) ed estrarre il componente appena programmato. Quest'ultimo é pronto per essere montato e provato sull'hardware dell'utente.

TERMINI DELLA GARANZIA

La garanzia copre il programmatore e tutte le sue parti, purchè esenti da difetti di fabbricazione, i materiali e la mano d'opera fino ad **un anno** dalla data di acquisto. La garanzia è altresì limitata a 25000 cicli per lo zoccolo ZIF DIL ed a 10000 cicli per glizoccoli ZIF non DIL. Se il prodotto viene diagnosticato come difettoso, il centro autorizzato di riparazioni riparerà o sostituirà le parti difettose gratuitamente. Le parti sostituite e/o l'intero programmatore saranno coperti solo per il rimanente pediodo di garanzia.

La garanzia non copre danni prodotti da usura, lacerazioni o danneggiamenti meccanici. Allo stesso modo non copre prodotti manipolati e/o riparati e/o aperti da personale non autorizzato dalla **grifo®**, o prodotti che sono stati danneggiati durante un trasporto oppure usati e/o installati e/o maneggiati impropriamente.

Prima di inviare il materiale in riparazione, si prega di contattare la **grifo®**.
In ogni caso, se il materiale giunge senza descrizione del problema la riparazione può essere respinta. Analogamente, non si può garantire la riparazione se non vengono inviati anche tutti gli elementi del programmatore (cavi di comunicazione ed alimentazione, POD diagnostici e gli eventuali accessori esterni) usati quando è stata riscontrata l'anomalia, una breve ma completa descrizione del difetto ed il report dei problemi.
Richieste di riparazioni non necessarie o fuori garanzia verranno addebitate.
La **grifo®** od i suoi distributori determineranno se il prodotto difettoso debba essere riparato o sostituito e giudicheranno se le condizioni di garanzia sono da applicarsi oppure no.

Si veda inoltre il capitolo RISOLUZIONE DEI PROBLEMI.

ARTICOLI AGGIUNTIVI

Per estendere con facilità la già vasta quantità di dispositivi programmabili con i programmatori **grifo®**, i convertitori di formato permettono di usare formati diversi dal classico DIL come il SDIP, PLCC, JLCC, SOIC, SOP, PSOP, SSOP, TSOP, TSOPII, TSSOP, QFP, PQFP, TQFP, VQFP, QFN (MLF), SON, BGA, EBGA, FBGA, VFBGA, UBGA, FTBGA, LAP, CSP, SCSP ecc.
Per ulteriori informazioni su quale convertitore di formato sia adatto al vostro programmatore e dispositivo si prega di usare la finestra visualizzata dal comando **Componente | Informazioni componente...** (<**Ctrl+F1**>) o di contattare direttamente la **grifo®**.
Da non dimenticare che il programmatore inoltre può utilizzare anche degli adattatori universali o speciali prodotti dalle case madri dei dispositivi senza doverne acquistare uno nuovo. Anche in questo caso consultare la finestra informativa visualizzata dal programma di controllo.

RISOLUZIONE DEI PROBLEMI

Il prodotto ed il materiale impiegato per costruirlo sono di ottima qualità. Nonostante ciò, se doveste incontrare dei problemi, rileggete attentamente la documentazione contenuta in questo manuale, probabilmente vi troverete la risposta cercata. Se così non fosse, seguite le istruzioni qui sotto.
Se non trovate una soluzione, potete contattare la **grifo®** direttamente. Prima si deve creare e spedire (via FAX o e-mail) un report diagnostico che manifesti il problema e poi si può procedere con un contatto telefonico. Preparatevi a dare una descrizione precisa della configurazione del vostro PC ed una descrizione del problema che vi è capitato. Non va tralasciato il tipo di PC, il produttore, la velocità, il sistema operativo e la versione, programmi residenti e la configurazione della porta di comunicazione usata.

PROLEMI DI LETTURA O PROGRAMMAZIONE

- Assicuratevi che il dispositivo target sia allineato correttamente nello zoccolo. Consultate la serigrafia accanto allo zoccolo o sull'adattatore se ne state usando uno.

- Il dispositivo target deve essere selezionato su PG4UW prima di tentare di leggerlo o programmarlo.

- Il nome del dispositivo selezionato ed il nome del dispositivo inserito nello zoccolo devono essere **ESATTAMENTE GLI STESSI!** Se tentate di operare su un dispositivo dal nome diverso o anche solo simile potreste danneggiare sia il programmatore sia il dispositivo.

- Aggiornate il programma di controllo. I produttori di elettronica creano nuovi dispositivi quasi ogni giorno per cui la cosa migliore è cercare il dispositivo che serve nell'ultima versione del PG4UW. Se non lo trovate consultate il paragrafo DISPOSITIVI NON SUPPORTATI.

- Se il dispositivo è vergine o è stato usato e poi cancellato si ritrova privo di contenuto. Effettuate un controllo tramite il menu **Componente | Verifica** cancellazione o premendo il tasto <**F6**> per accertarvi che il dispositivo sia leggibile e vuoto.

- **UEP 50** può effettuare un test di auto diagnostica per verificare il corretto funzionamento dei componenti interni. Seguire le indicazioni del paragrafo TEST DIAGNOSTICO (SELFTEST).

- Alcuni dispositivi proteggono il contenuto delle loro memorie tramite fusibili di protezione. Per poter gestire questi dispositivi i fusibili vanno disabilitati, usando le opzioni specifiche in **Componente | Opzioni componente | Opzioni operative**. Altrimenti l'utilizzo di questi dispositivi è impossibile, anche se i dispositivi, il PC ed il programmatore funzionano correttamente.

- Se la lettura e/o la programmazione avvengono tramite l'interfaccia ISP verificare che la scheda utente sia correttamente alimentata, eventualmente misurando tale tensione con un tester, direttamente sul dispositivo usato.

ERRORI DI COMUNICAZIONE

- Il programmatore deve essere alimentato correttamente, il LED verde di alimentazione deve essere sempre acceso e ad alta luminosità, l'alimentatore ed il cavo devono essere quelli forniti in dotazione. Per verificare l'integrità fisica del cavo di alimentazione provate a piegarlo leggermente dove si collega al programmatore e vedete se succede qualcosa.

- Aggiornate il programma di cotrollo, spesso una versione più recente migliora le caratteristiche di comunicazione. Consultate il capitolo AGGIORNAMENTO PROGRAMMA DI CONTROLLO per ulteriori informazioni.

- Il cavo di comunicazione USB deve essere quello fornito nella confezione.

- Provate ad installare il programmatore ed il PG4UW su un altro PC. Se questi funzionano bene sull'altro computer potreste avere dei problemi col primo. Confrontate le differenze tra questi due.

- A causa della grande varietà di porte parallele esistenti, può capitare che il programmatore non riesca a comunicare con il PC. Il problema può manifestarsi come comunicazione tra PC e programmatore indaffidabile o del tutto assente.

- Se si verifica questo comportamento, provate a collegare il programmatore ad altre porte parallele o ad altri PC.

- Verificate anche la presenza della porta collegata a livello del sistema operativo Windows, usando l'apposita finestra Avvio | Impostazioni | Pannello di controllo | Sistema | Hardware | Gestione periferiche | Porte (COM e LPT).

- Il programmatore va collegato direttamente alla porta parallela, non ci devono essere data switch nè chiavi hardware nè replicatori di porte tra PC e programmatore.

- La porta parallela va dedicata totalmente al programmatore, sotto Windows non va condivisa in rete, nè usata per una stampante od altre periferiche (scanner, plotter, ecc.)

- Il cavo di comunicazione parallela deve avere **tutti i piedini collegati.** Per verificare l'integrità fisica del cavo di comunicazione provate a piegarlo leggermente dove si collega al programmatore mentre PG4UW tenta di riconoscere il programmatore e vedete se succede qualcosa.

- Ricordate che la porta parallela alla quale è connesso il programmatore **deve** essere configurata come compatibile IEEE 1284. Per verificare la configurazione il modo migliore è entrare nel BIOS del vostro computer e vedere se la porta è impostata come ECP/EPP. Se usate una scheda multi I/O, assicuratevi che sia configurata come compatibile IEEE 1284.

- Nel caso la modalità ottimale descritta al punto precedente non funzioni, provate a configurare la porta parallela come compatibile Centronics (o standard, o normale o SPP: a seconda del produttore del BIOS che state usando si trovano nomi diversi per indicare la stessa cosa). Questo portebbe diminuire le prestazione ma assicurare la massima affidabilità nella comunicazione.

DISPOSITIVI NON SUPPORTATI

Dovendo usare dispositivi non supportati dal programma di controllo, leggete i consigli seguenti:

- Controllate l'elenco dei dispositivi supportati nell'ultima versione del programma di controllo disponibile dai nostri siti internet. Il dispositivo che vi interessa può essere già incluso nel database dell'ultima versione.

- Contattate la **grifo®** direttamente. Potremmo avere bisogno dei data sheets del componente e, possibilmente, dei campioni. I campioni verrano restituiti dopo che il dispositivo da voi richiesto sarà stato incluso nella prossima versione del PG4UW.

APPENDICE A: INDICE ANALITICO

A

AGGIORNAMENTO **21**
AIUTO **52**
ARTICOLI AGGIUNTIVI **57**
ASSISTENZA **1**, **57**

B

BINARIO **13**
BOCCOLE **11**, **23**
BUFFER **12**, **36**

C

CANCELLAZIONE **47**
CHECKSUM **38**
COMANDI **33**
  AIUTO **52**
  BUFFER **36**
  COMPONENTE **39**
  FILE **33**
  OPZIONI **50**
  PROGRAMMATORE **47**
COME INIZIARE **55**
COMPONENTE **39**
COMUNICAZIONE **59**
CONFIGURAZIONE PC **15**
CONNESSIONE CON PC **18**, **20**
CONNETTORI **11**, **26**
CONSUMO **8**
CONVERTITORI **57**

D

DESCRIZIONE SOFTWARE **30**
DIMENSIONI **8**
DIRETTIVE **1**
DISPOSITIVI **13**, **60**
DISPOSITIVO TARGET **3**, **13**

E

ELEMENTI **11**
ENCRYPTION **35**
EXORMAX **13**

F

FILE **13**, **33**

G

GARANZIA **57**

H

HANDLER **48**
HEX **3**, **13**

I

ID **41**
IEEE 1284 **59**
INFORMAZIONI DISPOSITIVO **47**
INSTALLAZIONE **16**
INTRODUZIONE **1**
ISP **3**, **10**, **24**

J

JEDEC **13**
JTAG **27**

L

LED **11**, **18**, **22**, **27**
LETTURA **46**
LINGUA **51**
LPT **15**, **20**

M

MARCHI REGISTRATI **2**
MISO **26**
MODALITÀ PROTETTA **51**
MOS **13**
MOSI **26**
MOTOROLA S-RECORD **13**
MULTIPROGRAMMAZIONE **22**

N

NORMATIVE **1**

O

OPERAZIONI **12**

OPZIONI **50**
OPZIONI COMPONENTE **41**
OPZIONI GENERALI **50**

P

PASSWORD **52**
PESO **8**
PG4UW **30**
POD DIAGNOSTICO **23**, **49**, **50**
PRECAUZIONI **21**, **24**
PREPARAZIONE PROGRAMMATORE **18**
PROBLEMI **58**
PROGETTO **33**
PROGRAMMA **47**
PROGRAMMA DI CONTROLLO **30**
PROGRAMMATORE **47**

R

REPORT **53**
REPORT DIAGNOSTICO **53**
RESET **26**
RISOLUZIONE PROBLEMI **58**
  DISPOSITIVI NON SUPPORTATI **60**
  ERRORI COMUNICAZIONE **59**

S

SCHERMATA PRINCIPALE **31**
SELEZIONE DISPOSITIVO **39**
SERIALIZZAZIONE **42**
SICUREZZA **1**, **12**
SOFTWARE **30**
STATISTICHE **45**
SUONO **51**

T

TASTI SPECIALI **32**
TEKTRONIX **13**
TEMPERATURA **8**
TEMPI **10**
TENSIONE DI LAVORO **8**
TEST DIAGNOSTICO **23**, **49**, **50**
TESTER DI CIRCUITI INTEGRATI **15**

U

UMIDITÀ **8**

USB **15**, **18**
USO **16**, **30**

V

VERIFICA **46**

Y

YES! **11**, **28**, **49**

Z

ZIF **3**, **13**

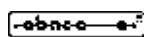
UEP 50

Universal Eprom Programmer - 48 pins, USB

USER MANUAL

UEP 50 is a small, fast and powerful programmer for all the programmable devices, available on the market.
Moreover, trough the **ISP** connector, it is capable to program directly the devices when they are already mounted on the user targer board.
The communication interface and the autonomous execution of the operations by the internal electronics, reduce at maximum either the data transfer time and the total programming time.
Its design and development tecniques allow an easy and fast addition of new devices, it has a remarkable low price, in addition to an high reliablity. It offers one of the best price/performances ratio in this range of products.

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661

UEP 50 Rel. 3.00 *Ed. 19 October 2012*

IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsability for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware.
grifo® reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product.
For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

Attention: Generic danger

Attention: High voltage

Attention: ESD sensitive device

Trade Marks

GENERAL INDEX

INTRODUCTION 1

PRELIMINARY INFORMATION 3
CONVENTIONS AND TERMINOLOGY 3
CONVENTIONS 3
TERMINOLOGY 3

GENERAL FEATURES 4

TECHNICAL FEATURES 8
PACKAGE INCLUDED 8
GENERAL 8
DACS SOCKET AND PINDRIVERS 8
ISP INTERFACE 10
PROGRAMMING SPEED 10
UEP50 ELEMENTS 11
DEVICE OPERATIONS 12
STANDARD 12
SECURITY 12
SPECIAL 12
BUFFER OPERATIONS 12
SUPPORTED FILE FORMATS 13
DEVICES SUPPORT 13
PARALLEL PROGRAMMING ON ZIF SOCKET 13
SERIAL PROGRAMMING THROUGH ISP INTERFACE 14
INTEGRATED CIRCUIT TESTER 15
MINIMAL PC CONFIGURATION 15
RECOMMENDED PC CONFIGURATION 15

INSTALLATION AND USE 16
CONTROL PROGRAM INSTALLATION 16
PROGRAMMER ARRANGEMENT 18
USB CONNECTION: DRIVER INSTALLATION 18
LPT CONNECTION 20
NEW VERSIONS OF CONTROL PROGRAM 21
USER CAUTION 21
LEDS 22
MULTIPROGRAMMING 22
DIAGNOSTIC SELFTEST 23
GROUND AND ESD BANANA JACK 23
ISP: IN SYSTEM PROGRAMMING 24
CAUTIONS FOR ISP PROGRAMMING 24
USE OF ISP PROGRAMMING 25
ISP CONNECTOR 26

SOFTWARE DESCRIPTION 30
RUN THE CONTROL PROGRAM 30
MAIN WINDOW 31
HOT KEYS 32
CONTROL PROGRAM COMMANDS 33
FILE 33
BUFFER 36
DEVICE 39
PROGRAMMER 47
OPTIONS 50
HELP 52
SOFTWARE ADDITIONAL FEATURES 54

HOW TO START 55

WARRANTY TERMS 57

ADDITIONAL TOOLS 57

TROUBLESHOOTING 58
READING OR PROGRAMMING PROBLEMS 58
COMMUNICATION ERRORS 59
UNSUPPORTED TARGET DEVICE 60

APPENDIX A: ALPHABETICAL INDEX A-1

FIGURE INDEX

FIGURE 1: COMPLETE VIEW OF UEP 50 5
FIGURE 2: PACKAGE CONTENT 9
FIGURE 3: PROGRAMMING TIMES 10
FIGURE 4: LEDS, CONNECTORS, BUTTON, ETC. LOCATION 11
FIGURE 5: MINI CD MANAGEMENT WINDOW 16
FIGURE 6: SUMMARY OF INSTALLATION SELECTIONS 17
FIGURE 7: USB DRIVER INSTALLATION (1 OF 3) 18
FIGURE 8: USB DRIVER INSTALLATION (2 OF 3) 19
FIGURE 9: USB DRIVER INSTALLATION (3 OF 3) 19
FIGURE 10: LEDS DESCRIPTION 22
FIGURE 11: SELFTEST PLUS POD 23
FIGURE 12: SELFTEST ISP CONNECTOR POD 23
FIGURE 13: CONNECTION FOR ISP PROGRAMMING 24
FIGURE 14: ISP CONNECTOR 26
FIGURE 15: ISP SIGNALS CONNECTION ON PROGRAMMER 27
FIGURE 16: ISP SIGNALS 28
FIGURE 17: PG4UW MAIN WINDOW 31

INTRODUCTION

The use of these devices has turned - IN EXCLUSIVE WAY - to specialized personnel.

This device is not a **safe component** as defined in directive **98-37/CE**.

Some componentes of the programmer are sensitive to ESD noises: so personnel who handles the product or the used devices is invited to take all necessary precautions that avoid possible damages caused by electrostatic discharges.

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

The informations for the installation, the assemblage, the dismantlement, the handling, the adjustment, the reparation, the addition of accessories, devices, installation, etc. are destined - and then executable - always and in exclusive way from specialized warned and educated personnel, or directly from the AUTHORIZED TECHNICAL ASSISTANCE, in the height respect of the manufacturer recommendations and the actual safety and health norms.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the Web sites www.grifo.com www.grifo.it or directly the AUTHORIZED TECHNICAL ASSISTANCE.

To prevent problems during product utilization, it is a good practice to read carefully all the informations of this manual, BEFORE STARTING ANY OPERATION. After this reading, the user can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult the on line help on the enclosed distribution disc for the last minute changes.

grifo® assumes no responsibility for misuse of this manual and it reserves the right to make changes or improvements to the product described in this manual at any time, without notice.

grifo® provid this documentation "as is" without warranty of any kind. In no event shall **grifo®** be liable for indirect, special, incidental or consequential damages of any kind arising from any error in this documentation, including any loss or interruption of business, profits, use , or data. Moreover is not guaranteed the updating of the product for new computers or new operating systems, that will become available in the future.

PRELIMINARY INFORMATION

UEP 50 is a real universal programmer and an integrated circuit tester, provided of 48 powerfull pin driver with many capabilities and functionalities. Its design allows to easily add new devices to the device list, to mantain a low price equal to a nice “value for money” in his class, to obtain an excellent price/performances ratio and at the same time it reach a high reliability of the programmation.
The availability of an ISP interface and of the two alternative communication interfaces (serial USB or parallel LPT) increases its application fields and simplifies the use.

CONVENTIONS AND TERMINOLOGY

There are some special conventions and terms used in this manual:

CONVENTIONS

References to the control program functions are in capitals, e.g. **LOAD**, **FILE**, **DEVICE**, etc. References to control keys are written in brackets <>, e.g. **<F1>**.

TERMINOLOGY

Target device Any kind of programmable integrated circuits or programmable devices.

ZIF socket Zero Insertion Force socket used for insertion of target device.

PC A Personal Computer necessary for the user to interact with the programmer.

Buffer Part of memory or disk, used for temporary data storage.

Porta USB Kind of high speed serial interface of PC, which is primarily dedicated to external peripherals connection as printer,removable disks, scanner, modem, digital photocamera, etc.

ISP In System Programming, that is the ability to program a device without removing it from its application board, even if it is soldered. The board must be compatible with this feature.

HEX format Format of data file, which may be read with standard text viewers; for examplebyte 5AH is stored as characters '5' and 'A', which means bytes 35H and 41H. One line of this HEX file (named record) contains start address, an identifier, data bytes and finally a security checksum.

DIL Package format with Dual In Line pins placed on the longer sides of the target device, for traditional mounting.

PG4UW Control program executed by PC that manages the programmer.

GENERAL FEATURES

This chapter contains a short loquacios description of the programmer, that includes a summary of its technical and functional features. All the detailed information are described in the following chapters.

UEP 50 is a first member of next generation of USB/LPT compatible, Windows based **grifo® universal programmers** built to meet the strong demand of the small manufacturing and developers community for the fast and reliable universal programmer, capable to manage all the devices on the market.

UEP 50 supports all kinds of silicon technologies of today and tomorrow programmable devices without family specific module. You have freedom to choose the optimal device for your design. Using built in In System Programming (**ISP**) connector,the programmer is able to program Icompatible chips in circuit.

UEP 50 isn't only programmer, but also **tester** of TTL/CMOS logic integrated circuits and memories. Furthermore, it allows generating user definable test pattern sequences, in order to test even special or composed devices.

UEP 50 provides very competitive price coupled with excellent hardware design for reliable programming. Probably **best "value for money"** programmer in this class.

UEP 50 provides **very fast programming** due to high speed FPGA driven hardware and execution of time critical routines inside of the programmer. At least fast than competitors in this category, for many chips much faster than most competitors. As a result, when used in production this one socket programmer waits for an operator, and not the other way round.

UEP 50 interfaces with the IBM PC Pentium compatible or higher, portable or desktop personal computers through USB (1.1, 2.0).

UEP 50 provides a banana jack for ESD wrist straps connection in order to easy implement the ESD protection control and also other banana jack for earth wire.

UEP 50 has a FPGA based totally reconfigurable 48 powerful TTL pindrivers, where provide H/L/pull_up/pull_down and read capability for each pin of socket. Our advanced pindrivers incorporate **high quality** and **high speed** circuitry that deliver signals without overshoot or ground bounce for all supported devices. Improved pindrivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low voltage devices.

The programmer performs device **insertion test** (wrong or backward position in socket) and **contact check** (poor contact pin to socket) before it programs each device. These capabilities, supported by **overcurrent protection** and **signature byte check** help prevent chip damage due to operator error. Through some accessories and the supplied control program, the **UEP 50** is capable to execute a **diagnostic self test**, that checks automatically the pindrivers, the voltage levels, the timings and the

communication with PC. In other words this feature checks the health of the programmer.

Built in **protection circuits** eliminate damage of programmer and/or programmed device due environment or operator failure. All the inputs of the programmer, including the ZIF socket, connection to PC, ISP interface and power supply input, are strongly **protected against ESD** up to 15 KV.

UEP 50 programmer performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention.

FIGURE 1: COMPLETE VIEW OF UEP 50

UEP 50 is driven by an **easy to use** control program with pull down menu, hot keys and on line help. Selecting of device is performed by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. **Standard** device related commands (read, blank check, program, verify, erase) are enhanced by some **test functions** (insertion test, signature byte check), and some special functions (autoincrement, production mode, start immediately after insertion of chip into socket)). The control program allows the manipulation of the device data temporaneously saved inside a proper **buffer**, the transfert of buffer from/to file on disk, the automatic file format detection and conversion during load of file.

The rich featured **autoincrement function** enables to assign individual serial numbers to each programmed device, or simply increments a serial number; alternatively the function enables to read serial numbers and/or identification signatures of any programmed device, from a file.

The software also provides a many information about programmed device. As a special, the **drawings of all available packages**, explanation of **chip labeling** (the meaning of prefixes and suffixes at the chips) for each supported chip are provided.

Control program also features a full set of information about programmed device. For example, in the information section of a specific device, after it has been selected, it is possible to find the **drawings of all available packages** and the explanation of **chip labeling** that is the meaning of prefixes and suffixes on the chip names, for each supported device.

Furthermore the software provide a full information about ISP implementation: description of ISP connector pins for currently selected chip, recommended board design around the in system programmed chip, the user modality and other necessary information.

The **remote control** feature allows the software flow controlled by other application; this happen either by using .BAT commands file or using DLL library. Some examples (in C, PASCAL, VBASIC, .NET) and proper manual are part of standard software delivery.

Jam files of JEDEC standard JESD-71 are interpreted by **Jam Player**. Jam files are generated by design software which is provided by manufacturer of respective programmable devices and they include all specifications about chips programming on ZIF socket or through ISP connector (IEEE 1149.1 of Joint Test Action Group = JTAG interface).

VME files are interpreted by VME Player. VME file is a compressed binary variation of SVF file and contains high level bus operations. VME files are generated by design software which is provided by manufacturer of respective programmable devices.and they include all specifications about chips programming on ZIF socket or through ISP connector (IEEE 1149.1 of Joint Test Action Group = JTAG interface).

Multiple devices are possible to program and test via **JTAG interface** with the two association just described: ISP-JAM or ISP-VME.

By attaching more **UEP 50** programmers to the same PC (through USB ports) it is achieved a **powerful multiprogramming system**, which **support as many chips, as those supported by connected programmer** and without obvious decreasing of **programming speed**. It is mportant to know, there is a concurrent multiprogramming and each programmer works independently from the others; so each programmer can program either the same or different chip, if necessary.

The high speed, the multiprogramming and the status signals for external systems make of **UEP 50** the right system even for **production department**, in fact it is the fovourite one (for his flexibility and performances), even when it is compared with multisocket programmers (gang programmers).

It is important to remember that in most cases new devices require **only a software update** due to the **UEP 50** is truly universal programmer. With our prompt service you can have new devices added to the current list within few days!

Various **socket converters** are available to handle device in PLCC, SOIC, PSOP, SSOP, TSOP, TSSOP, TQFP, QFN (MLF), SDIP, BGA, LAP, CSP and other **packages**. The control program recognizes the format upon the complete device name and, when required, it informs with the name and order code of the adapter module.

PACKAGE INCLUDED

- Programmer **UEP 50**.
- USB cable for connecting the programmer to PC.
- ISP cable (see ISP: IN SYSTEM PROGRAMMING paragraph).
- Power supply cable for mains voltage.
- Diagnostic POD for selftest of programmer's ZIF socket.
- Diagnostic POD for selftest of programmer's ISP connector.
- Anti dust cover for ZIF socket.
- Mini CD with the control program, this User’s manual and additional utility files.
- Programmer calibration test report.
- Carton shipping case.

These components can be checked on figure 2 that shows all them except the last two items.

GENERAL

- Power supply voltage: 110 - 240 Vac, 50 - 60 Hz.
- Power consumption max. 20 W when completely active, about 2 W in sleep mode.
- USB 2.0 high speed communication port, up to 480 MBit/s transfer rate, USB 1.1 compatible.
- On board powerful microprocessor and FPGA based state machine.
- Banana jack for ESD wrist straps connection.
- Banana jack for connection to ground.
- Protection against surge and ESD on power supply input and communication ports.
- Dimensions 197 x 140 x 56 mm (H x W x D).
- Metallic container with rubber feets.
- Weight ca. 1100 g (programmer alone); 1550 g (complete package).
- Temperature 5 - 40 °C
- Humidity 20 - 80 %, non condensing.

DACS SOCKET AND PINDRIVERS

- Three D/A converters for VCCP, VPP1 and VPP2, with controllable rise and fall time.
- VCCP range 0 - 8 V; 1 A.
- VPP1 and VPP2 range 0 - 26 V; 1 A.
- Selftest capability.
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins;
- Pindriver: 48 universal drivers capable to:
    - connect each pin to VCCP/VPP1/VPP2;
    - perfectly connect each pin to ground;
    - connect each pin to TTL driver.
- Analog pindriver output level selectable from 1.8 V up to 26V.
- Current limitation, overcurrent shutdown, power failure shutdown.
- 48 pins DIL ZIF (Zero Insertion Force) socket accepts both 300, 600 mil devices up to 48 pins.
- ESD protection on each pin of socket according with IEC1000-4-2 standard: 15 KV on air, 8 KV on contact.
- Continuity test: each pin is tested before every programming operation.

FIGURE 2: PACKAGE CONTENT

ISP INTERFACE

- 20 pins, male type with missinsertion lock connector.
- Six TTL driver for ISP generate H, L, CLK, pull-up and pull-down; output analog level H settable from 1.8 V up to 5 V to handle all devices, including new low voltage devices.
- One VCCP voltage for ISP (range 2÷7 V, 100 mA).
- The VCCP for ISP has both source/sink capability and voltage sense.
- One VPP voltage for ISP (range 2÷25 V, 50 mA).
- One power supply voltage for target board (range 2V÷6 V, 250 mA).
- ESD protection on each pin of ISP connector according with IEC1000-4-2 standard: 15 KV on air, 8 KV on contact.
- Two output digital signals, which indicate state of work result: LEDOK and LEDERROR (active level: min 1.8 V).
- One digital input signal, switch YES! equivalent, that proceed with next operation (active level: max 0.8 V).

PROGRAMMING SPEED

These times strongly depend on selected communication connection but not on PC configuration. In the following tables are reported some times measured with a PC *Pentium IV; 2.4 GHz; USB 2.0 HS; Windows XP*.

DEVICE, TYPE DIMENSION	OPERATIONS	MODE	TIME
M50FW080, parallel Flash 100000Hx8=8 MBit=1 MByte	programming and verification	on ZIF socket	22 sec
MX28F640C3BT, parallel Flash 400000Hx16=64 MBit=8 MByte	programming and verification	on ZIF socket	57 sec
K9F1G08U0M, par. NAND Flash 8400000Hx8=1 GBit=128 MByte	programming and verification	on ZIF socket	239 sec
AT45D081, serial Flash 108000Hx8=8 MBit=1 MByte	programming and verification	on ZIF socket	36 sec
AT89C51RD2, microcontroller 10000Hx8=512 KBit=64 KByte	programming and verification	on ZIF socket	15 sec
PIC18LF452, microcontroller 4000Hx16=256 KBit=32 KByte	programming and verification	on ZIF socket	4 sec

FIGURE 3: PROGRAMMING TIMES

UEP 50 ELEMENTS

The programmer is composed by the following elements directly manageable by the user:

1) YES! button
2) LED for complete power/sleep mode (POWER)
3) Power supply connector
4) 40 pins ZIF socket
5) LED for operation under execution (BUSY)
6) USB connector for communication with PC
7) ISP connector (see ISP: IN SYSTEM PROGRAMMING paragraph)
8) LPT connector (Option)
9) LED for operation correctly executed (GOOD)
10) Banana jack for ground connection
11) LED for operation executed with problem (ERROR)
12) On/Off switch
13) Banana jack for ESD wrist straps connection.

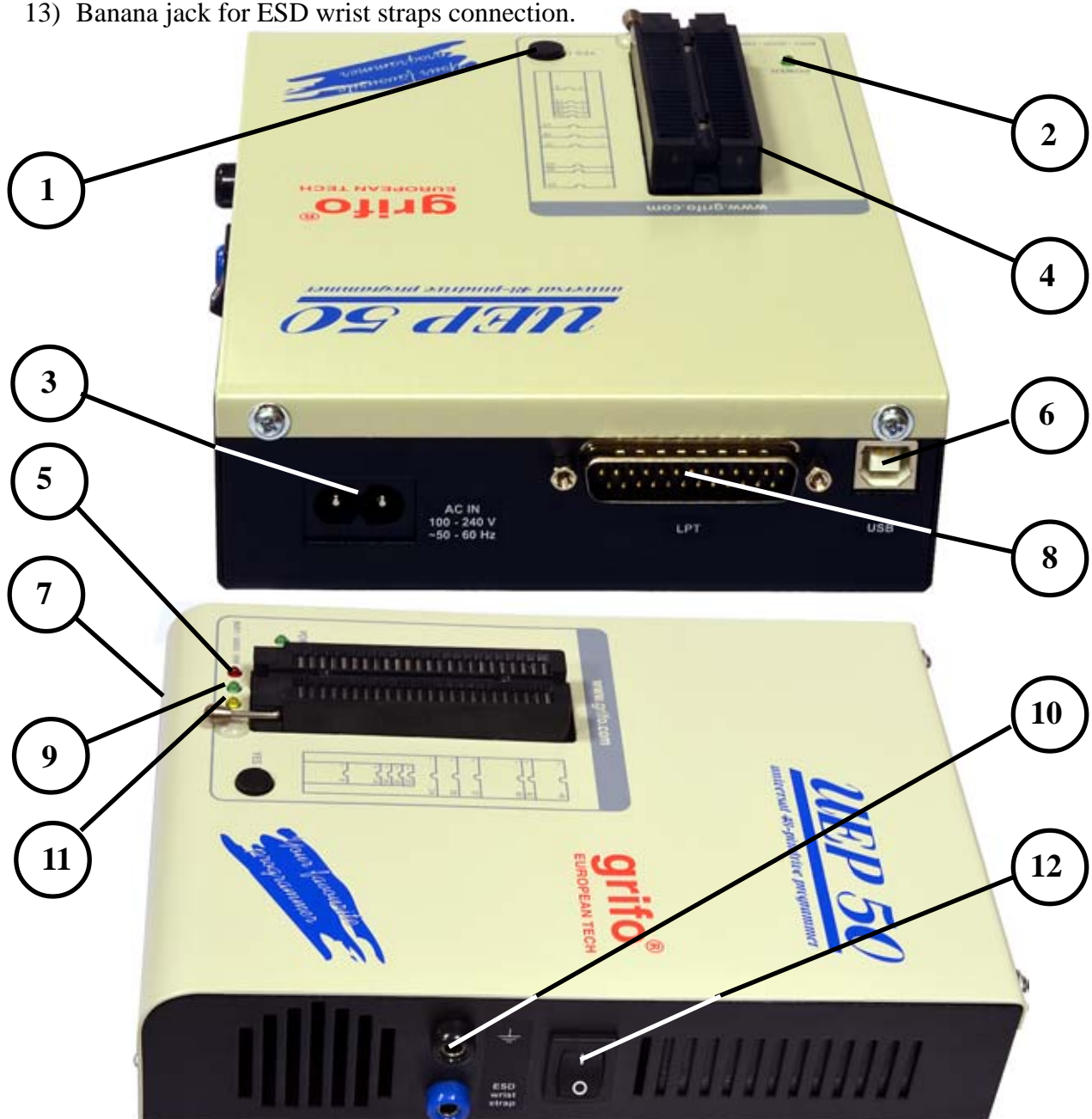



FIGURE 4: LEDS, CONNECTORS, BUTTON, ETC. LOCATION

DEVICE OPERATIONS

When the **UEP 50** programmer is used with his control program executed on PC, it can execute the following groups of operations. When the used device is provided of some internal memories of different types, the following operations can be performed ob all these memory types.

STANDARD

- Intelligent device selection by device type, manufacturer or typed fragment of part name.
- Automatic selection of EPROM, Flash EPROM, microcontrollers., based on internal IDentifier (signature)
- Blank check, read and verify to check programmation results.
- Program.
- Erase.
- Illegal bit test.
- Checksum.

SECURITY

- Insertion test of device.
- Electric contact check.
- ID and signature byte check.
- Management of protection fuses.

SPECIAL

- Statistic.
- Auto serialization of device with different modes: simple incremental mode, from file mode, custom generator mode.
- Count down mode for programmed device.
- Management of configuration fuses.
- Interpreter of JAM files and of Standard Test and Programming Language (STAPL) according with JEDEC standard JESD-71 for programming and test.
- Interpreter of VME files compressed binary variation of SVF (Serial Vector Format) files.
- Production mode with automatic start immediately after the device insertion.

BUFFER OPERATIONS

In this paragraph are described all the operations that the programmer can perform on the buffer that contains the data read/write from/on the used device.

- View/edit, find/replace.
- Fill/copy, move, byte swap, word or double word split.
- Checksum (in many different standard format).
- Print.
- File load/save.
- No download time because programmer is directly controlled by PC.
- Automatic file type identification.

SUPPORTED FILE FORMATS

The control program of **UEP 50** is capable to manage autonomously the following standard file formats:

- Unformatted (raw) binary.
- HEX Intel, HEX Intel extended.
- Motorola S-record.
- MOS.
- Exormax.
- Tektronix.
- ASCII-space-HEX.
- ASCII HEX.
- JEDEC (ver. 3.0.A), produced by most important development tools as: ABEL, CUPL, PALASM, WARP, TANGO PLD, OrCAD PLD, PLD Designer ISDATA etc.
- JAM (JEDEC STAPL Format), JBC (Jam STAPL Byte Code), STAPL (STAPL File) according with JEDEC standard JESD-71.
- VME (ISP VME file version VME2.0 and VME3.0).

DEVICES SUPPORT

Below are briefly described the main gruops of devices families supported by **UEP 50**. A complete list of these devices, updated and detailed, can be obtained through the control program and his search modalities.

PARALLEL PROGRAMMING ON ZIF SOCKET

- EPROM: NMOS/CMOS, 2708 (*), 27xxx and 27Cxxx series, with 8/16 bit data width and full support of Low Voltage models.
- EEPROM: NMOS/CMOS, 28xxx, 28Cxxx, 27EExxx series, with 8/16 bit data width and full support of Low Voltage models.
- Flash EPROM: 28Fxxx, 29Cxxx, 29Fxxx, 29BVxxx, 29LVxxx, 29Wxxx, 49Fxxx series, from 256 KBit to 1 GBit, with 8/16 bit data width and full support for Low Voltage models.
- Serial EEPROM: 24Cxxx, 24Fxxx, 25Cxxx, 45Dxxx, 59Cxxx, 25Fxxx, 25Pxxx, 85xxx, 93Cxxx, NVM3060, MDAxxx series, full support for Low Voltage models.
- Configuration EEPROM: XCFxxx, XC17xxxx, XC18Vxxx, EPCxxx, AT17xxx, 37LVxx included the Low Voltage models.
- One-Wire EEPROM: DS1xxx, DS2xxx.
- PROM: AMD, Harris, National, Philips/Signetics, Tesla, TI.
- NV RAM: Dallas DSxxx, SGS/Inmos MKxxx, SIMTEK STKxxx, XICOR 2xxx, ZMD U63x series.
- PLD Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX7000AE, MAX II series.
- PLD Lattice: ispGAL22V10x, ispLSI1xxx, ispLSI1xxxEA, ispLSI2xxx, ispLSI2xxxA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, LC4xxxB/C/V/ZC, M4-xx/xx, M4A3-xx/xx, M4A5-xx/xx, M4LV-xx/xx series.
- PLD Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II.

- PLD by other manufacturer: SPLD/CPLD series: AMI, Atmel, AMD-Vantis, Gould, Cypress, ICT, Lattice, NS, Philips, STM, VLSI, TI.
- Microcontrollers 48 family: 87x41, 87x42, 87x48, 87x49, 87x50 series.
- Microcontrollers 51 family: 87xx, 87Cxxx, 87LVxx, 89Cxxx, 89Sxxx, 89LVxxx by all manufacturers, Philips and Nexpedia LPC series.
- Microcontrollers Intel 196 series: 87C196 KB/KC/KD/KT/KR/... series.
- Microcontrollers Atmel AVR: AT90Sxxxx, ATtiny, ATmega series.
- Microcontrollers Cypress: CY7Cxxxxx, CY8Cxxxxx series.
- Microcontrollers ELAN: EM78Pxxx serie.
- Microcontrollers MDT: 1xxx and 2xxx series.
- Microcontrollers Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17Cxxx, PIC18xxx, PIC24xxx, dsPIC series.
- Microcontrollers Motorola (Freescale): 68HC05, 68HC08, 68HC11, HCS08, HCS12 series.
- Microcontrollers Myson: MTV2xx, 3xx, 4xx and 5xx series.
- Microcontrollers National: COP8xxx serie.
- Microcontrollers NEC: uPD78Fxxx serie.
- Microcontrollers Novatek: NT68xxx serie.
- Microcontrollers Renesas: M3xxx, HD64Fxxxx series.
- Microcontrollers Scenix (Ubicom): SXxxx serie.
- Microcontrollers SGS-Thomson: ST6xx, ST7xx, ST10xx, STR7xx series.
- Microcontrollers TI: MSP430 and MSC121x series.
- Microcontrollers ZILOG: Z86/Z89xxx and Z8xxx series.
- Microcontrollers by other companies: EM Microelectronic, Fujitsu, Goal Semiconductor, Hitachi, Holtek, Princeton, Macronix, Winbond, Infineon(Siemens), Samsung, Toshiba, etc.

SERIAL PROGRAMMING THROUGH ISP INTERFACE

- Serial EEPROM: models with I2C, MicroWire, SPI interfaces and other models as KEELOQ series, serial data Flash, PLD configuration memories.
- Microcontrollers Atmel: AT89Sxxx, AT90Sxxxx, ATtiny, ATmega series.
- Microcontrollers Cypress: CY8C2xxxx serie.
- Microcontrollers Elan: EM78Pxxx, EM6xxx series.
- Microcontrollers EM Microelectronic: 4 and 8 bit series.
- Microcontrollers Microchip PICmicro: PIC10xxx, PIC12xxx, PIC16xxx, PIC17xxx, PIC18xxx, PIC24xxx, dsPIC series.
- Microcontrollers Motorola/Freescale: HC11 series, HC908 series (both 5-wires, All-wires), HCS08, HCS12 series.
- Microcontrollers NEC: uPD7xxx serie.
- Microcontrollers Philips, Nexpedia: LPC2xxx, LPC, 89xxx series.
- Microcontrollers Scenix (Ubicom): SXxxx serie.
- Microcontrollers TI: MSP430 (both JTAG and BSL), MSC12xxx series.
- PLD Lattice: ispGAL22xV10x, ispLSI1xxxEA, ispLSI2xxxE, ispLSI2xxxV, ispLSI2xxxVE, ispLSI2xxxVL, M4-xx/xx, M4LV-xx/xx, M4A3-xx/xx, M4A5-xx/xx, LC4xxxB/C/V/ZC series.
- Various PLD (also by JAM player with JTAG support):
  Altera: MAX 3000A, MAX 7000A, MAX 7000B, MAX 7000S, MAX 9000, MAX II
  Xilinx: XC9500, XC9500XL, XC9500XV, CoolRunner XPLA3, CoolRunner-II

INTEGRATED CIRCUIT TESTER

- TTL gates: 54,74 S/LS/ALS/H/HC/HCT series.
- CMOS gates: 4000, 4500 series.
- Static RAM: 6116÷624000.
- User definable test pattern generation.

NOTE:

(*) Obsolete device, programming with additional module.

MINIMAL PC CONFIGURATION

Below are listed the minimum requirements of the PC that is connected and used together with the programmer; these requirements are valid for the current version of control program and they can change according with its value. The user can obtain updated requirements inside the on line help of **PG4UW**.

- Pentium III.
- 128 MB free RAM.
- Hard Disk drive with 60 MB free space.
- One CD drive.
- One communication port: USB 1.1 (or higher) or LPT (ECP/EPP).
- Colour monitor with 800 x 600 pixel.
- Mouse and keyboard.
- Operating system: MS Windows 2000/XP.

RECOMMENDED PC CONFIGURATION

Free hard disk space requirements depends also on used IC device size. For large devices the required free space on disk will be approximately 60MB + device size.

- Pentium IV.
- 512 MB free RAM.
- Hard Disk drive with 150 MB free space.
- One CD drive.
- One USB 2.0 High Speed communication port.
- Colour monitor with 1024 x 768 pixel.
- Mouse and keyboard.
- Operating system: MS Windows XP.

INSTALLATION AND USE

In this chapter are reported the most important information about programmer activation, complete of some suggestions for the user that simplify and and ensures the right use in any operative condition.
It is strongly recommended to execute the operations described in the paragraphs of this chapter in the exact order and to install the software before connecting programmer to PC.
Furthermore the user can take advantages by reading the next chapters: the first (SOFTWARE DESCRIPTION) describes the control program in details and the second (HOW TO START) lists the step required to perform the first programming.

CONTROL PROGRAM INSTALLATION

1) Select the PC that must be used with the programmer and check that it cover the minimal requirements described in previous chapter.
2) If there is already a PG4UW control program installed on selected PC, first of all uninstall it. To obtain this, from the Windows Start menu, select "Program", then Grifo(r) and finally "Uninstall Pg4uw" or use classic uninstall modalities of Windows operating systems.
3) The programmer package contains a mini CD with the control program, the user manuals of all programmers and the Acrobat Reader program useful for reading these documents.
   Insert delivered mini CD to your CD drive and instal program starts automatically by showing the following window. Whenever the window doesn't appear, please run manually the "SETUP.EXE" program, saved on the same CD.

FIGURE 5: MINI CD MANAGEMENT WINDOW

4) Press the button "Software installation PROGRAMMERS" and wait that the installation program is executed.

5) Select the English language that will be used during the installation and press the "Ok" button.
6) Read the displayed information windows then, as suggested, close all the other possible programs in execution and then press the button "Next".
7) Selects the folder on the hard disk of PC where the control program must be installed and confirm with "Next" button. At this time the installation program advise about the possible presence of the selected folder and the user can decide if it must be used or not.
8) Selects the folder with the execution link of the program, that must be added at the "Start" menu of Windows and press the "Next" button.
9) Selects how many and which icons for fast launch must be added on PC (as for example those on the desktop, those on Windows bar, etc.).
   Moreover select if the support for the control of Multiprogramming must be installed, as described in following paragraph MULTIPROGRAMMING.
   At this point press the "Next" button.
10) Check that int the displayed summarizing window all the performed selections are corrected and thus confirm with the "Install" button.

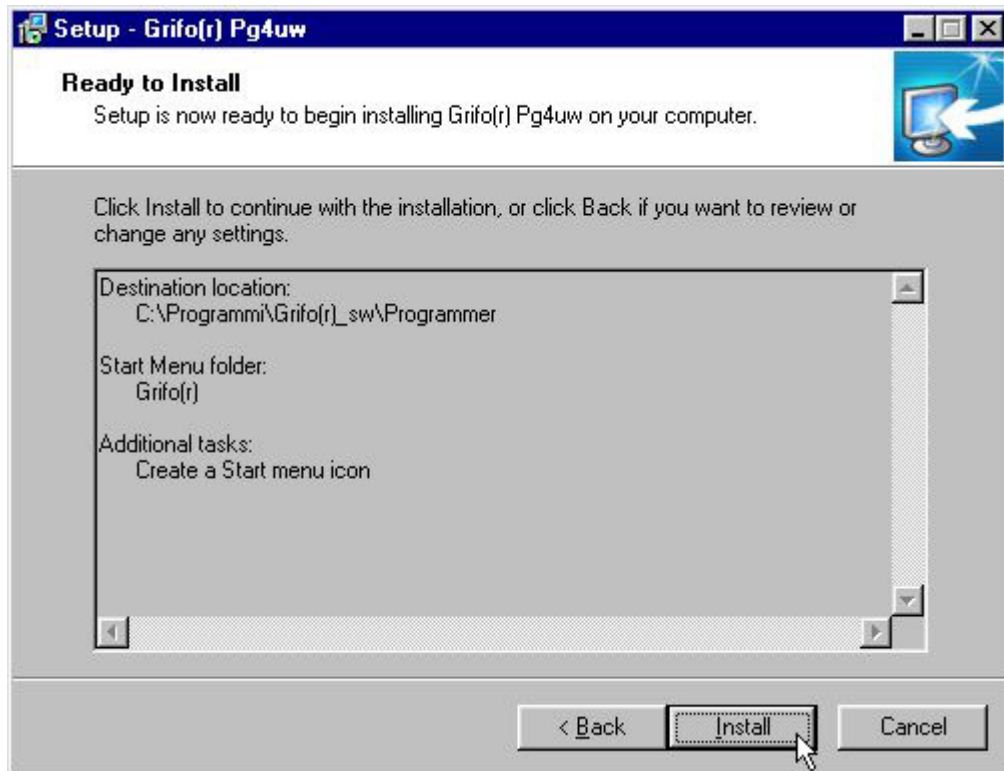



FIGURE 6: SUMMARY OF INSTALLATION SELECTIONS

11) Wait the end of installation indicated by a proper progressive bar.
12) At this point according with used PC and operating system, it could be displayed a window that requires the restart of PC: select "Yes, restart computer now". In any way press the "End" button to close definitely the installation program and to return at the window of figure 5.
13) In this phase all the software required by the programmer has been copied on hard disk of PC. The user manual can be read on PC through the Acrobat Reader program; if the last is not already installed it can be done by pushing the "Install Adobe Acrobat Reader" button.
14) Exit from installation program by pushing in sequence the "Exit" and "Close" buttons.
15) In order to obtain a comfortable environment it is suggested to copy the user manual, in PDF format, from \Documentation\ folder of mini CD to hard disk of PC.

PROGRAMMER ARRANGEMENT

16) Verify that all the elements of the programmer (described in previous paragraph PACKAGE INCLUDED) are available inside the carton shipping box.
17) Connects the power supply cable to a 230 Vac mains voltage and to programmer.
18) Turn the lateral switch in on position (I) and verify that immediately after the LED POWER is turned on with high brightness, then the other LEDs BUSY, GOOD, ERROR are turned on and off in sequence and finally that only the LED power remains on, at low brightness. This one is the normal power on sequence when PC is not connected.

At this point the required operations change according with selected communication interface. The following two paragraphs describe these operations either for serial USB interface and LPT parallel interface.

USB CONNECTION: DRIVER INSTALLATION

As for any peripheral device of PC, connected through the USB port, also for the **UEP 50** the operating system must have the proper software drivers, that allows its management. These drivers are copied during the installation of the control program, thus in order to avoid useless complications, it is strongly suggested to perform before the operation described at points 4÷18 and only after to connect the programmer.
The following steps are the right ones under Windows XP, but the installation procedure is really similar under the other Windows operating systems.

19) Connect the programmer to PC by using the USB cable provided in the received box.
20) At this point the PC automatically executes the "Found new hardware wizard" composed by some sequential phases as described in following steps. During these phases the mini CD inserted at point 3 must be still present in the PC drive.
21) In the first phase select “Install the software automatically (Recommended)", as illustraded in following figure, and click “Next” to continue the installation.

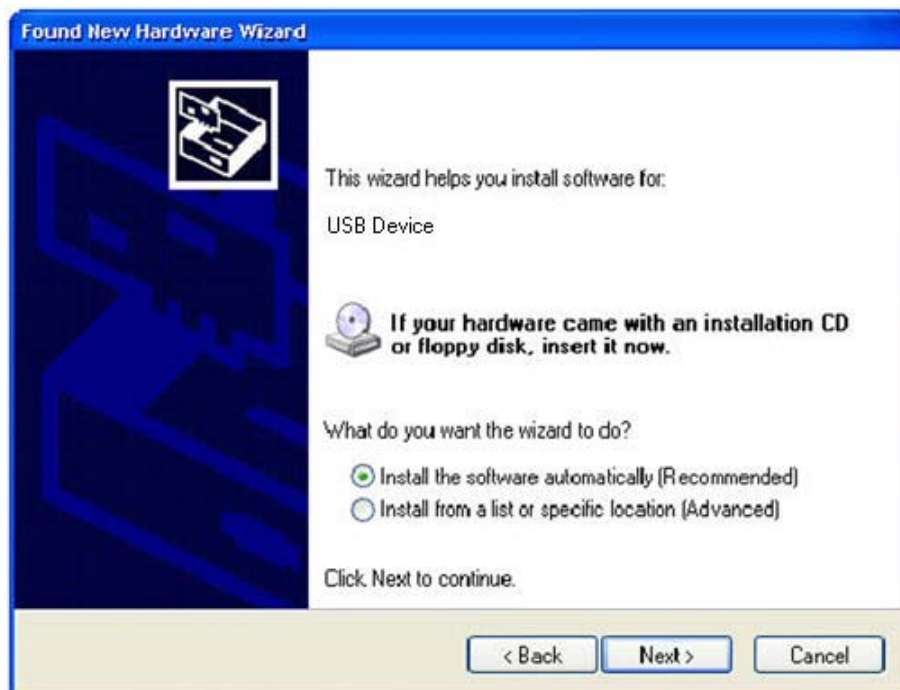



FIGURE 7: USB DRIVER INSTALLATION (1 OF 3)

22) Immediately after the PC will start the research of the best drivers for the new hardware on all its disks and it will find them on the supplied mini CD. At this point it is necessary to continue the software installation even if it doesn't respect the Windows compatibility, by pushing the "Continue Anyway" button.

FIGURE 8: USB DRIVER INSTALLATION (2 OF 3)

23) Atr the end of driver installation and registration, check if the new added device works correctly by comparing the shown window with the following figure: then close the window with the "Finish" button.

FIGURE 9: USB DRIVER INSTALLATION (3 OF 3)

24) At this point the USB driver installation is complete and the programmer is ready to work. Please remind that next reconnections and/or power ons of programmer will not require the driver, in fact they are already available, if the used USB port is the same. Viceversa when a different USB port is used, it will be necessary to execute again the steps 20÷23.

LPT CONNECTION (OPTION)

As an alternative to USB serial connection, the **UEP 50** can be connected to LPT parallel interface of PC. This interface on PC side is those dedicated to printer management, according with Centronics or IEEE1284 standards.
Also for this connection type in order to avoid useless complications it is suggested before to perform the installation described at points 4÷18 and then execute the following steps.

19) Switch off PC and programmer.
20) Get a cable for PC parallel port that is a 1:1 extension with D 25 pins connectors (male and female) provided of shielding conductor. The user can acquire this communication cable from computer accessories shops.
21) Insert the communication cable to a free printer port on your PC. If your computer is equipped with only one printer port, substitute the printer cable with the programmer cable. Connect the opposite cable edge to **UEP 50** and screw the anchorings on both the connectors. This is very important in order to avoid randomic malfunctions.
    It may be uncomfortable to switch between printer cable and programmer cable, but it is not possible to operate the programmer through a mechanical or an electronic data switches. However you can install a second multi I/O card in your computer, thus obtaining a supplementary printer port, say LPT2. So your printer may remain on LPT1 while the programmer on LPT2.
22) Turn on the PC.
23) Turn on the programmer and check that the activation sequence for the LEDs, already described at point 18, is re-executed.
24) At this time the **UEP 50** is ready to work..

! CAUTION !

If You don't want to switch off your PC when connecting/disconnecting the programmer, proceed as follows:

- When connecting the programmer to the PC: **FIRST** insert the LPT communications cable and **THEN** supply power to programmer (or switch it on).
- When disconnecting the programmer from the PC: **FIRST** disconnect the power supply from programmer connector (or switch it off) and **THEN** remove the LPT communication cable.

From programmer's point of view the connecting and disconnecting sequence is irrelevant. Protection circuits of the programmer inputs keep it safe. **But think of your PC, please**.

Whenever the communication between PC and programmer described in next chapter SOFTWARE DESCRIPTION, is not correctly established please re-check the previous steps and if the problem remain, consult the TROUBLESHOOTING chapter at the end of manual.

NEW VERSIONS OF CONTROL PROGRAM

Semiconductor manufacturers continuously introduce new devices with new package types, manufactured by new technologies in order to support the need for flexibility, quality and speed in product design and manufacturing, as the market requires.
Furthermore, a typical programmable device undergoes several changes during its lifetime in an effort to maintain or to improve its technical characteristics and process yields. These changes often impact with the programming algorithms, which need to be upgraded (the programming algorithm is a set of instructions that tells the programmer how to program data into a particular target device). Using the newest algorithms in the programming process is the key to obtaining high quality results. In many cases, while the older algorithm will still program the device, they may not provide the level of data retention that would be possible with an optimal algorithm. Failure to not use the most current algorithm can decrease your programming yields (more improper programmed target devices), and may often increase programming times, or even affect the long term reliability of the programmed device.
At least, we are making mistakes too and some bugs can bypass all the performed tests.

For the described reason **grifo®** periodically develops a new version of control program that solve all the described problems. For example, we usually implement more than 500 new devices into the control program within a year and tens new algorithms are added. In order to exploit all the capabilities of programmer we recommend using the latest version of PG4UW.EXE. To obtain this version the customer must:

- Uninstall the previous version by using proper link "Windows Start | Program | Grifo(r) | Uninstall Pg4uw" and follow the shown instructions. Whenever the uninstall process advise about the removing of drivers, please confirm the operation taking care that they are used only by PG4UW.
- Download from internet web site: **http://www.grifo.com**
http://www.grifo.it
the latest version of control program, that is the file "PG4UWARC.EXE".
- Run the dowloaded file that is the installation program of the new version.
- At this point it is possible to install the new control program by executing the steps 5÷23 described in previous paragraphs.

We suggest to check the release of new versions about one time per month or when the same control program advise about possible new version, when it is executed.

USER CAUTION

The internal electronics of **UEP 50** protect the target device and the programmer itself against many use anomaly. However, it is not possible to grant the integrity of the two systems in each use conditions, and thus the customer must respect the following cautions:
- target device can be damaged by incorrect selections or user selected programming parameters.
- target device may be not destroyed by forced interruption of the control program (PC reset or switched off), by removing the physical connection to the programmer. If the device is not damaged the content of actually programmed cell may remain undefined;
- don’t unplug the target device from the ZIF socket during work with devices (LED BUSY shine);
- a wrong position of the device on the ZIF socket can produce a damage or a complete breakage of the component;
- electrostatic dicharges cause unpredictable results on each electronic devices;
- respect all the cautions described in ISP: IN SYSTEM PROGRAMMING paragraph.

LEDS

UEP 50 is provided with 4 LEDs to visualize power supply status and operating programmer status. Please refer to figure 4 to locate the LEDs.

LED NAME	COLOUR	DESCRIPTION
POWER	Green	It signals the power supply status of the programmer between two possible conditions: high brightness when power supply is complete; low brightness when power supply is reduced (sleep).
BUSY	Red	When on, it indicates that the programmer is accessing the target device, both for reading or writing. Please do not touch the target device when this LED is on.
GOOD	Green	When on, it indicates that the last executed operation has been performed successfully.
ERROR	Yellow	When on, it indicates that an error condition occoured during the last executed operation.

FIGURE 10: LEDS DESCRIPTION

NOTE:
When the **UEP 50** programmer isn't active, control program switch the programmer into "sleep mode" and it consumes less than 2 Watt of power; this current can be completely reduced by using the proper On/Off switch, on the side of programmer.
When programmer is connected to power supply, you will see the three LEDs BUSY, GOOD, ERROR turning on and then switching off in sequence, in order to let the user check their functionality.

MULTIPROGRAMMING

The **UEP 50** support the multiprogramming that is the contemporaneous use of more programmers, connected on USB, with a single PC. The following points briefly describe the most important steps that must be performed in order to use this feature:

- During installation, in the Additional Tasks window, choose to install the multiprogramming control support (see point 9 of CONTROL PROGRAM INSTALLATION).
- Connect all the programmers and perform the installation of USB drivers.
- Run special control program PG4UWMC.EXE and select the used programmer.
- Load configuration projects for each connected programmer.
- Run PG4UW control program for every connected and assigned programmer.

The multiprogramming is useful especially when production of many devices (of the same and different types) is required. It is normally used by two different operators: the qualified technician that arranges the project files and the unqualified user that use the programmer already configured. For details and other information please read carefully the on line documentation of the PG4UWMC program, in fact it coordinates all the multiprogramming.

DIAGNOSTIC SELFTEST

If you feel that your programmer does not react according to your expectation, please run the programmer selftest by using proper diagnostic PODs (for ZIF socket and ISP connector), enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test in order to prevent possible anomalies that can produce unwanted effects on the programmed devices.
See instructions for selftest execution in the proper paragraphs PROGRAMMER | SELFTEST, PROGRAMMER | SELFTEST PLUS... and PROGRAMMER | SELFTEST ISP CONNECTOR... in SOFTWARE DESCRIPTION chapter and relative on line help.
Remind that the first command must be executed with an **empty ZIF socket or the programmer could be damaged**. The second must be executed with the diagnostic POD for ZIF socket inserted:

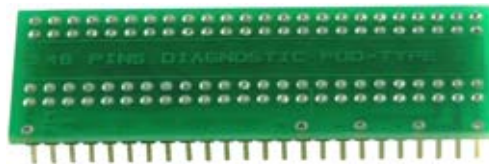



FIGURE 11: SELFTEST PLUS POD

And finally the third command must be executed with the diagnostic POD for ISP inserted in the ZIF socket and connected with ISP cable to ISP connector of programmer:

FIGURE 12: SELFTEST ISP CONNECTOR POD

For optimal results with programmer we recommend you also periodic executions of calibration test, through the specific **Programmer | Calibration test...** command of control program. This check require a dedicated 48 pins calibration test POD, that can be directly acquired from grifo®.

GROUND AND ESD BANANA JACK

On the right side of programmer there are two banana jack with the following functions:

Ground banana jack: it can be connected to ground in order to ensure a perfect balancing of the electric reference signals of the interconnected systems; for example it is really useful when ISP programming is used.

ESD banana jack: it can be connected to protections against electro static doscharges, used by the operator that manage the programmer and the devices. Among these protections we remind the wrist straps, collars, ankles and any other suitable accessory.

The banana jack position on programmer can be easily found by using figure 4.

ISP: IN SYSTEM PROGRAMMING

In System Programming allows programming and reprogramming of device already mounted on the user board or inside the final system. Using a simple interface, the ISP programmer communicates serially with the device, reprogramming non volatile memories of the chip.
In system programming eliminates the physical removal of chips from the user board and this will save time, and money, both during development in the lab, and when updating the software or parameters in the field.
Moreover the ISP interface can be used also to read the contents of all the internal memories of target device, obtaining the possibility to get from the board a list of parameters, data, configurations either set during previous installation or collected by the same card during its normal work.

CAUTIONS FOR ISP PROGRAMMING

We recommend respect following rules to avoid damages to PC, ISP programmer, target device or target system:

- Ensure **common ground point** for user board whhere the target device is mounted, ISP programmer and PC.
- For laptop or other PC which is not connected to common ground point: make hard wired connection from laptop to common earth point (for example use one of the **metallic D connector** chassis **of the PC**).
- Any other devices connected to user board connected to programmer must be connected to common ground point too.

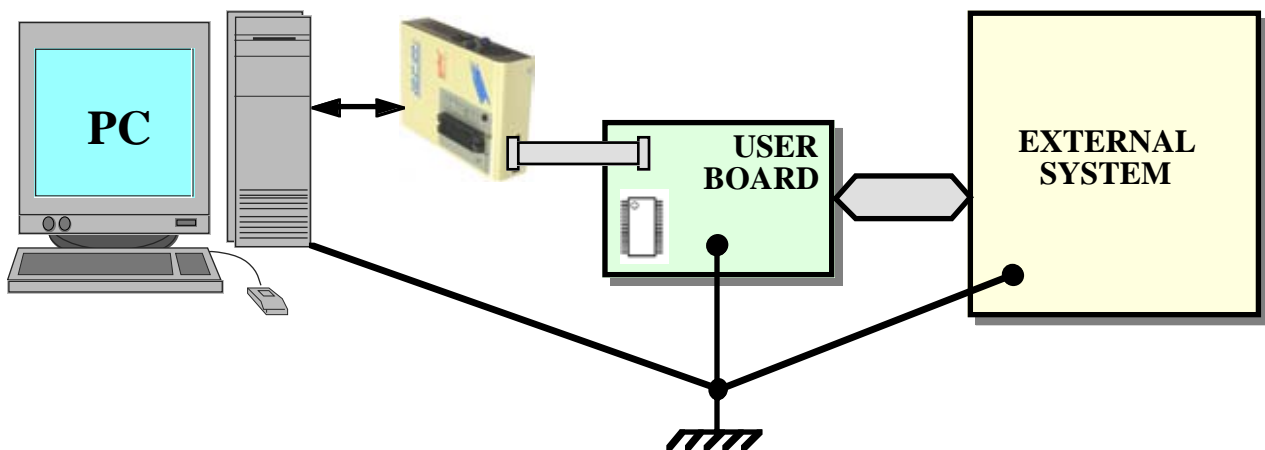



FIGURE 13: CONNECTION FOR ISP PROGRAMMING

- When you use **UEP 50** as ISP programmer, **don’t insert device into ZIF socket**: it must be empty.
- When you programm devices in ZIF sosket of **UEP 50**, **don’t insert ISP cable to ISP connector**.
- Use only **attached ISP cable**. When you use other ISP cable (other material, length…), programming may occour unreliable.

- **UEP 50 can** supply power to target device but with a limited current: please check carefully the respect of these limit but, in order to be extremely sure, we suggest to provide a dedicated supply voltage to user board.
- User board with target device **can't** supply power to **UEP 50:** please connect programmer power supply cable and turn it on with switch on the right side.
- **UEP 50** tests supply voltage of target device and checks his value at the beginning of any operation to execute. If the programming voltage is different as expected, no action with target device will be executed.
- During in system programming you must connect two electrical devices (ISP programmer and target system) according with indications of control program on PC . Unqualified connection can damage these devices.
- If the user board uses signals needed for ISP programming for other functions and circuitry, it’s necessary separate them from target device by appropriate resistors or jumpers. In other words the user board must be developed in order to support the ISP programming (see following descriptions).

NOTE: ***When you don’t keep below directions and you damage programmer during in system programming, it is a damage caused by unqualified manipulation and is out of warranty***.

USE OF ISP PROGRAMMING

During in system programming you must connect two electrical devices (ISP programmer and target system) according with incications of control program on PC . Unqualified connection can damage these devices. After all the cautions described in previous paragraph have been satisfied, the user must:

1) Turn off both devices: ISP programmer and target device. To ensure this condition even when the programmer has its own ON/OFF switch, please free the power supply connector.
2) Assign same GND potential for all devices, e.g. connect GND of all devices by wires.
3) Insert one connector of ISP cable to ISP programmer, turn on programmer and run control program.
4) In control program select target device and operation options. During this selection please take care about the suffix ISP at the end of the device name.
5) Start required action on target device (read, program).
6) When instructions message of control program appears, connect other ISP cable connector to user board and turn it on. Once these steps have been done press the **Ok** button in order to proceed.
7) At this time the operation started at point 5 proceed until it is completed.
8) When instructions message of control program appears, disconnect ISP cable connector from user board and turn it off.
9) If you need another action on target device, or you must repeat the operation on another user board, please return to step 5.

ISP CONNECTOR

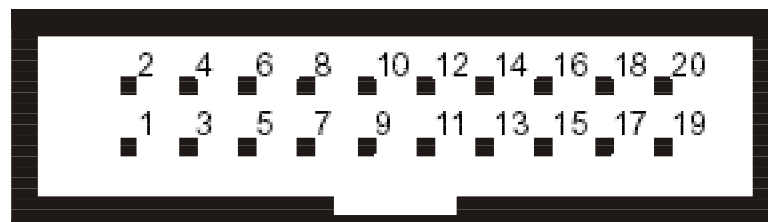
The programmer is provided with a low profile connector, male, 10 ways, with missinsertion lock for the the in system programming of devices that are already mounted on user boards. This connector has an easy access in fact it is on the down side of the programmmer, as described on figure 4; among the received items there is also an ISP cable for the connection of user board.
Specification of ISP connector pins depends on programming device and it is displayed on **Device info** window. The above mentioned window appears pressing <**Ctr +F1**> keys combination and contains several useful information, in detail in the tab **ISP connection details** it is possible to find the connector pin out and example connection diagram for the ISP target board.

Correspondance between ISP signals and connector pins <u>may vary according to the device to be programmed</u>.
To know this correspondance, please select the device desired in the control program (menu **Device | Select Device**) and use the **Device | Device Info** command or press <**Ctrl+F1**>.

FIGURE 14: ISP CONNECTOR

Once the ISP cable is connected to ISP connector, the <u>pin number 1 is signed by triangle scratch on both sides connectors</u>.

<u>The user that need the ISP programming must develop his board in order to be compatible with this modality</u>. in other words the user board must be designed to allow all signals, which are used for in system programming to be directly connected to ISP programmer via ISP connector. If target system use these signals for other functions, it is necessary to isolate these signals. Target system mustn't affect these signals during in system programming.
For in system programmable devices manufacturers publish application notes. Design of **grifo®** programmers together with respect of these application notes allow proper in system programming. Alternatively the customers can read the electric diagrams described always in the **ISP connection details** tab.

Please remind that on user board should respects the following suggestions:
- provide a diode to protect the target circuit against a higher voltage, which is provided by ISP programmer;
- if your target board supply differs from mentioned 5V, choose please use a Zener diode for protection, with a voltage suitable to this supply voltage;

- we recommend to use resistors to separate the target device from target system. If pins needed for ISP programming are inputs in target system then separation by resistors is sufficient and resistors make a low pass filter too. If pins are outputs, then use of resistors saves a programming time.
- of course the isolation resistors can be replaced by switches or jumpers, if necessary. In that case, before and after the ISP programming of target device the switches (jumpers) must be moved. But the using of switches (jumpers) adds a next manipulation time to programming procedure.
- When on ISP connector are used the JTAG signals, please refer to normative of this standard, that is the IEEE 1149.
- On ISP connector there are two output signals that inform about the status of performed operation and they can be used by other systems for production and/or test of the user board. These signals are designed for a direct link to a LED and consequently they must be connected to compatible signals.
- On ISP connector there is an input signal that proceed with the operation under execution and it can be used by other systems for production and/or test of the user board. This signal is designed for a direct link to a normally open push button, that short circuit the input to ground and consequently it must be connected to compatible signal.
- Further electric information about ISP signals are available in the previous ISP INTERFACE paragraph.

The internal circuit of **UEP 50** connected to ISP and JTAG signals, can be compared to the siplified circuit described in the next figure:

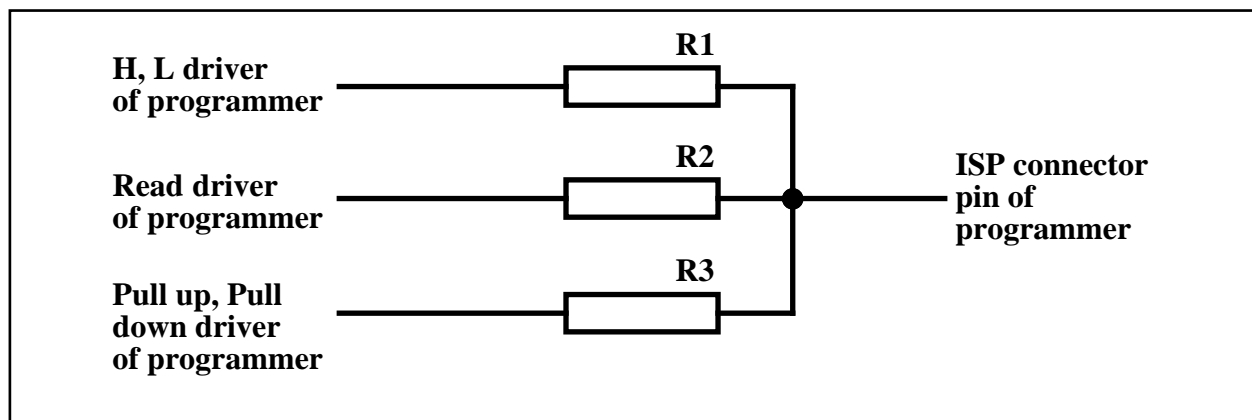



FIGURE 15: ISP SIGNALS CONNECTION ON PROGRAMMER

Whenever the documentation found about the ISP signals is not sufficient or completely absent, the user can use the following table that list the name and a brief description of the signals connected to ISP connector.

PIN	SIGNAL	PURPOSE
PGM	ProGraM	Low voltage (single supply) I SP programming enable pin.
PGC	ProGram Clock	ISP programming clock, generated by the in system programmer.
PGD	ProGram Data	ISP programming data (In/Out).
MCLR	Master CLeaR	Master Clear inpu is an active low Reset to the device. The in system programmer must control the target device reset signal.
Vpp	Programming Voltage	Programming voltage required for double supply device.
SCK	Serial Clock	Programming clock, generated by the in system programmer (master).
MOSI	Master Out Slave In	Communication line from in system programmer (master) to used target device (slave).
MISO	Master In Slave Out	Communication line from target device (slave) to in system programmer (master).
RESET	Target Reset	To enable in system programming, the target device RESET or /RESET must be kept active. To obtain this, the in system programmer must control the target device reset signal.
SDATA	Serial DATA	ISP programming data (In/Out).
TDI	Test Data Input	Communication line from programmer to target device, trough JTAG interface.
TDO	Test Data Output	Communication line from target device to programmer, trough JTAG interface.
TCK	Test CLocK	Programming clock Clock trough JTAG interface.
YES!	YES	Digital input for external YES push button (enabled when connected to ground), useful to start the operation through an external system.
LEDOK	OK LED	Digital output for external OK LED (enabled when at high level), useful to inform an external system about correct execution of last operation.
LED ERROR	ERROR LED	Digital output for external ERROR LED (enabled when at high level), useful to inform an external system about wrong execution of last operation.
Vdd	Power Supply voltage	The target and the in system programmer must be powered by their own source. Anyway the in system programmer test the target device power supply voltage, thus they must be connected.
GND (Vss)	Ground signal	The two systems (in system programmer and target device) must share the same common ground.

FIGURE 16: ISP SIGNALS

The previous table doesn't describe all the ISP signals of all the programmable devices by this modality, but only the most important ones that are largely diffused.

In order to obtain the complete description of ISP signals we remind the user to consult the technical documentation of the used device (data sheets and application notes); when these signals have been identified the user can obtain their connection with the ISP connector of the programmer through the **ISP connection details** tab of the **Device info** window, already described in previous pages.

As the infrormation windows displayed by control program change according with selected device, **it is really important that the user has correctly selected the name of the device he want to program with the suffix ISP at the end of device name**.

SOFTWARE DESCRIPTION

This chaptere contains an essential description of the control software **PG4UW.EXE**. Please remind that in previous chapter are described the installation modalities of the program and the connections modalities of the programmer; obviously this operations must be performed before using it. Moreover the next chapter HOW TO START shows a brief list of operations sufficient for the user that want to perform a first programmation.
Below are described some general notes about PG4UW:
- the on line help is updated before than the manual: the user must always use this comfortable tool and the included search modalities;
- program is common for all **grifo®** programmers.
- the program works correctly when executed on PC provided of the features described in MINIMAL PC CONFIGURATION paragraph;
- background operations under Windows is error free;
- the control program is granted to be free from any viruses at the moment of delivery. To increase safety our programs include a special algorithm for detecting possible virus infections.

RUN THE CONTROL PROGRAM

In order to execute PG4UW it can be selected the proper link in the Start menu of Windows (placed in the folder selected during the installation) or it can be used a double click to the icons added always during the installation phase.
When the PG4UW is started, program is checked for its integrity and if it is valid then it automatically scans all the existing ports of PC and it searches for a connected and powered **grifo®** programmer. Program PG4UW is common for all the **grifo®** programmers, hence program try to find all available models. Finally the program display the main window and waits for your instructions.
If the control program cannot communicate with the programmer, an error message appears on the screen, including error code and description of possible reasons (disconnected programmer, bad connection, power supply failure, incompatible USB or LPT port, programmer serial number not valid, etc). Eliminate the error source and press any key.
If error condition still exist, the program resumes its operation in the demo mode and access to the programmer is not possible. If you cannot find the cause of the error, follow the instructions in TROUBLESHOOTING chapter.
In addition, the control program checks communication with programmer prior to any operation with the programmed device.

MAIN WINDOW

In the main window of PG4UW program are always displayed the following area:

Header bar	name, copyright statement and version of PG4UW control program
Menu bar	list of basic functions
Statistics window	statistic information about programmer activity
Programmer window	information about the status of the programmer and PG4UW
Addresses window	organization, size, start and end addresses of the target device, of the buffer and used file
Device window	all relevant information about current target device
Help bar	a brief description of selected command

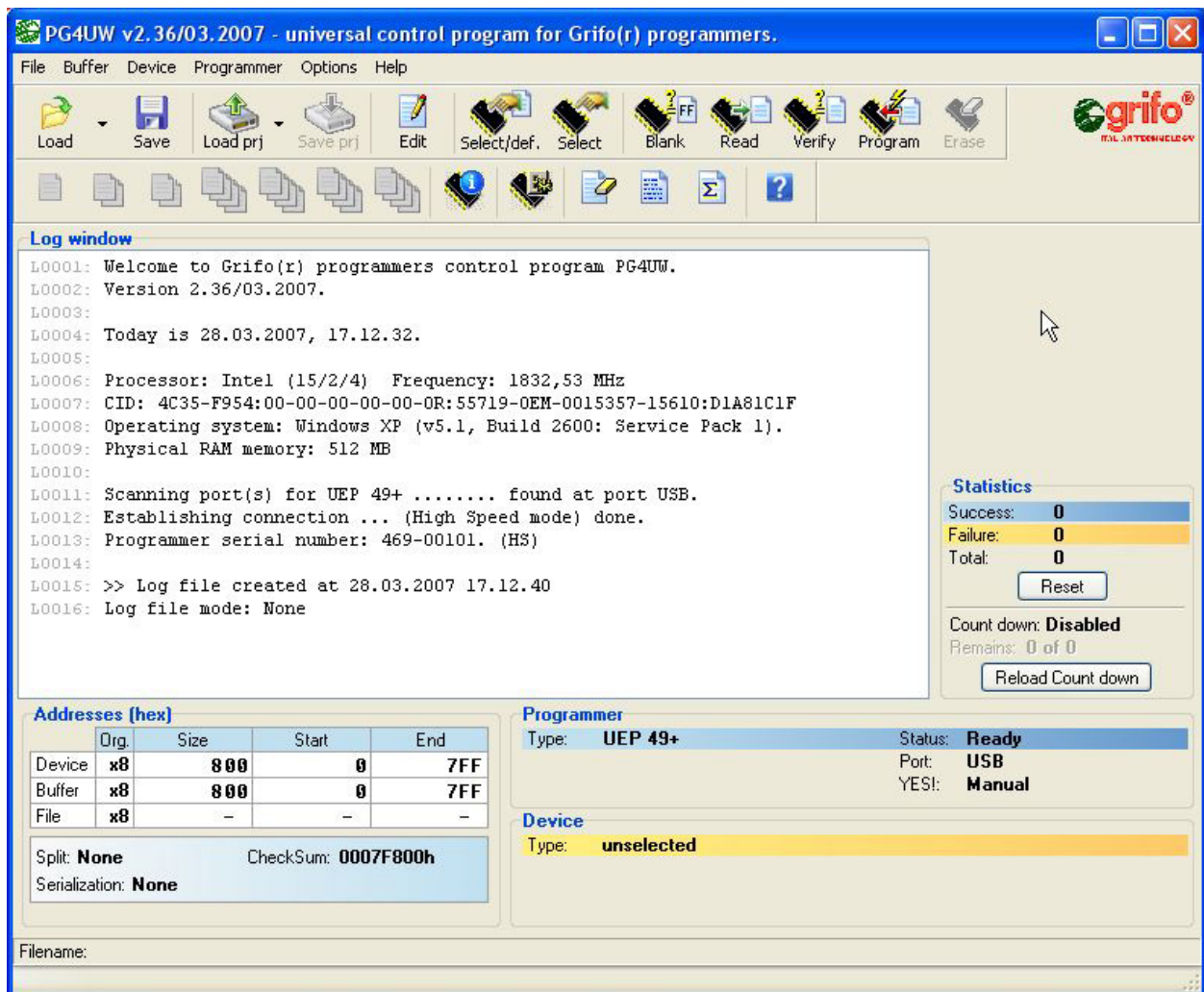



FIGURE 17: PG4UW MAIN WINDOW

Menu selection is carried out in the normal graphic user interface fashion, either by cursor moving plus pressing <**Enter**>, or by typing the highlighted letter in the wanted menu or - of course - by mouse.

! CAUTION !

All data entered through keyboard are in HEX format, except ASCII characters blocks in **Buffer | View/Edit** command.

HOT KEYS

PG4UW manages many hot keys that are available for quicker selection of the intensely used commands.

<**F1**>	Help	Calls Help
<**F2**>	Save	Save file
<**F3**>	Load	Load a file into the buffer
<**F4**>	View/Edit	Viewing and editing of buffer
<**F5**>	Select/default	Target device selection from 10 last selected devices list
<**Alt+F5**>	Select device	Target device selection by typing device or vendor name
<**Ctrl+F5**>	Select by ID	Target device selection by device family
<**F6**>	Blank check	Check if the device is completely erased
<**F7**>	Read	Reads device's content into the buffer
<**F8**>	Verify	Compares contents of the target device with the buffer
<**F9**>	Program	Programs target device
<**Alt+Q**>	Exit without save	Terminates the PG4UW
<**Alt+X**>	Exit and save	Terminates the PG4UW and saves settings too
<**Ctrl+F1**>	Device info	Displays additional information about current device
<**Ctrl+F2**>	Erase	Fills the buffer with a given value
<**Ctrl+Shift+F2**>	Fill random data	Fills the buffer with random values
<**Ctrl+F**>	Find programmer	Search for any programmer connected to PC
<**Shift+Ctrl+F**>	Refind programmer	Research for the programmer already connected to PC
<**Shift+Alt+D**>	Device list	Generates a list of all the devices managed by all the programmers

CONTROL PROGRAM COMMANDS

The following pages list all the commands available with control program, divided in groups according with main menus of the PG4UW. The command name is matched with is keyboard shortcut (if present) and a brief descripion of its purpose and employ modalities. Please remember that the most recently updated instuctions list for the control program is always the on line help, which should be consulted before this manual.

FILE

This submenu is used for source files manipulation, settings and viewing directory, changes drives, changes start and finish address of buffer for loading and saving files by **binary**, **MOTOROLA**, **MOS Technology**, **Intel (extended) HEX**, **Tektronix, ASCII space**, **JEDEC**, and **POF** format.

File | Load... <F3>

Analyse file format and loads the data from specified file to the buffer. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**). The control program stores a last valid mask for file listing. You can save the mask into the config. file by command **Options | Save options**.

Checking the check box **Automatic file format recognition** tells program to detect file format automatically. When program can't detect file format from one of supported formats, the binary file format is assumed.

When the check box **Automatic file format recognition** is unchecked program allows user to manually select wished file format from list of available file formats on panel **Selected file format**. When Binary file format is selected, there can be specified Buffer start value. Buffer start value is buffer address from which data read from file will be written to buffer.

The reserved key **<F3>** will bring out this menu from any menu and any time.

File | Save... <F2>

Saves data in the buffer, which has been created, modified, or read from a device onto a specified disk. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**).

The reserved key **<F2>** will bring out this menu from any menu and any time.

File | Load project...

This option is used for loading project file, which contains device configuration buffer data saved and user interface configuration.

The standard dialog **Load project** contains additional window - **Project description** - placed at the bottom of dialog. This window is for displaying information about currently selected project file in dialog Load project.

Project information consists of:
- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

File | Save project...

This option is used for saving project file, which contains settings of device configuration and buffer data saved. Data saved to project file can be restored anytime by menu command **File | Load project**.

The dialog **Save project** contains three additional windows in **Project description** panel placed at the bottom of dialog **Save project**. The windows are for displaying information about currently selected project file in dialog **Save project** and information about current project which has to be saved.

Project information consists of:
- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

The first (upper) window contains information about currently selected project file in dialog **Save project**.

The second (middle) windows displays information about actual program configuration including currently selected device, active programmer, date and time and program version. These actual program settings are used for creation of project description header.

The third (bottom) window is user editable and contains project description (arbitrary text) which usually consists of project author and some notes.

File | Reload file

Choose this option to reload a recently used file.

When you use a file, it is added to the **Reload file** list. Files are listed in order depending on time of use of them. Lastly used files are listed before files used far off.

To Reload a file:

1. From the File menu, choose Reload file.
2. List of lastly used files is displayed. Click the file you want to reload.

Note: *When reloading a file the file format is used, by which the file was lastly loaded/saved.*

File | Reload project

Choose this option to reload a recently used project.

When you use a project, it is added to the **Reload project** list. Projects are listed in order depending on time of use of them. Lastly used projects are listed before projects used far off.

To Reload a project:

1. From the File menu, choose Reload project.
2. List of lastly used projects is displayed. Click the project you want to reload.

File | Project options...

This option is used for display/edit project options of actually loaded project. Project options means basic description of project including following project data:

- device name and manufacturer
- project creation date
- version of program by which project was created
- user defined project description (arbitrary text), e.g. project author and other text data for more detailed project description

User can directly edit user defined project description only. Device name, manufacturer, project date and program version are generated automatically by program.

File | Load e table...

This command loads the data from binary file from disk and it saves them into the part of memory, reserved for an encryption (security) table.

File | Save e table...

This command writes the content of the memory's part, reserved for an encryption table, into the file on the disk as a binary data.

File | Exit without save...

The command deallocates heap, cancels buffer on disk (if exists) and returns back to the operation system.

File | Exit and save...

The command deallocates heap, cancels buffer on the disk (if exists), saves current setting of last 10 selected devices to disk and returns back to the operation system

BUFFER

This menu is used for buffer manipulation, block operation, filling a part of buffer with string, erasing, checksum and of course editing and viewing with other items (find and replace string, move data, printing, etc.).

Buffer | View/Edit... <F4>

This command is used to view (view mode) or edit (edit mode) data inside the buffer. Use arrow keys for select the object for edit. Edited data are signified by colour.
The following functions can be enabled by proper keys and menu, or alternatively by the buttons available in the View/Edit Buffer window.

Key	Function
<F1>	display help of actual window
<F2>	fill block causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.
<Ctrl+F2>	erase buffer with specified blank value
<Ctrl+Shift+F2>	fill buffer with random data
<F3>	copy block is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.
<F4>	move block is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.
<F5>	swap bytes command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).
<F6>	print buffer
<F7>	find string (max. length 16 ASCII characters)
<F8>	find and replace string (max. 16 ASCIIchars.)
<F9>	change current address
<F10>	change mode view / edit
<F11>	switch the mode of buffer data view between 8 bit and 16 bit view. It can be also do by mouse clicking on the button to the right of View/ Edit mode buffer indicator. This button indicates actual data view mode (8 bit or 16 bit), too.
<F12>	checksum dialog allows to count checksum of selected block of buffer change mode view / edit
<Arrow keys>	move cursor up, down, right and left
<Home/End>	jump on start / end current line
<PgUp/PgDn>	jump on previous / next page
<Ctrl+PgUp/PgDn>	jump on start / end current page
<Ctrl+Home/End>	jump on start / end current device
<Backspace>	move cursor one position left (back)

Note: *characters 20H - FFH (mode ASCII) and numbers 0..9, A..F (mode HEX) immediately changes content of edit area.*
Warning: *Editing of ASCII characters for word devices is disabled.*

Print buffer
This command allows write selected part of buffer to printer or to file. Program uses an external text editor in which selected block of buffer is displayed and can be printed or saved to file, too. By default is set simple text editor **Notepad.exe**, which is standard part of all versions of MS Windows.
In Print buffer dialog are following options:
Defines start address of selected block in buffer.
Block end
Defines end address of selected block in buffer.

Find dialog box
Enter the search **String to find** to text input box and choose **Find** to begin the search or choose **Cancel** to forget it.
Direction box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.
Origin specifies where the search should start.

Find & Replace dialog box
Enter the search string in the **Text to find** string input box and enter the replacement string in the **Replace with** input box.
In **Options** box you can select prompt on replace: if program finds instance you will be asked before program change it.
Origin specifies where the search should start.
Direction box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.
Press **<Esc>** or click **Cancel** button to close dialog window.
By pressing **Replace** button the dialog box is closed and a Question window is displayed. This window contains following choices:

Yes	replaces found item and finds next
No	finds next item without replacing current one
Replace All	replaces all found items
Abort search	aborts this command

Buffer | Fill block...

Selecting this command causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.

Buffer | Copy block...

This command is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.

Buffer | Move block...

This command is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.

Buffer | Swap data in block...

This command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).

Buffer | Erase... <Ctrl+F2>

If this command is selected, the content of the buffer will be filled with topical blank character. The reserved key **<Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer | Fill random data... <Shift+Ctrl+F2>

If this command is selected, the content of the buffer will be filled with random data. The reserved key **<Shift+Ctrl+F2>** will bring out this menu from any menu and any time.

Buffer | Duplicate buffer...

This command performs duplicate buffer content in range of source EPROM to range of destination EPROM. This procedure is suitable if there is used for example 27C512 EPROM to 27C256 EPROM position.

Note: *The procedure always uses buffer start address 00000h.*

Buffer | Checksum...

The checksum dialog is used for calculate checksums of selected block in buffer. The checksums are calculated by next way :

Byte	sum by bytes to “word”. CY flag is ignored
Word	sum by words to “word”. CY flag is ignored
Byte (CY)	sum by bytes to “word”. CY flag is added to result.
Word (CY)	sum by words to “word”. CY flag is added to result.
CRC-CCITT	sum by bytes to “word” using RESULT=PREVIOUS+(x^16 + x^12 + x^5 +1)
CRC-XModem	sum by bytes to “word” using RESULT=PREVIOUS+(x^16 + x^15 + x^2 +1)

Column marked as **Neg.** is a negation of checksum so, that Sum + Neg. = FFFFH.
Column marked as **Suppl.** is complement of checksum so, that Sum + Suppl. = 0 (+ carry).
Dialog checksum contains following items:

From address: This is a start address of block selected for calculating checksums in buffer. Address is defined as Byte address.

To address: This is an end address of block selected for calculating checksums in buffer. Address is defined as Byte address.

Insert checksum: This is special item used for select which kind of checksum will be written into the buffer when, the **Calculate & insert** was executed.

Insert address: This is special item that specifies an address from the buffer where a result of chosen checksum will be written, when the **Calculate & insert** was executed. Address can not be specified inside the range **From address** to **To address**. Address is defined as Byte address.

Size: This item is used for setting a size of chosen checksum result, which will be written into the buffer. A size of checksum result may be 8 (byte) or 16 (word) bits long. If word size was selected, whole checksum value will be written into the buffer. In other case will be written only low byte of checksum value.
Note: *If word size was selected, a low byte of checksum value will be written on address specified in box Insert address and a high byte will be written on address incremented by one.*

Calculate: Click on the button Calculate starts calculating checksums for selected block in buffer. No writes into the buffer are executed.

Calculate & insert: Click on the button **Calculate & insert** starts calculating checksums for selected block in the buffer and writes the chosen checksum into the buffer on address specified by **Insert address**.

Close: Click on this button and the Ckecksum window is closed.

DEVICE

The functions for a work with selected programmable devices - device select, read data from device, device blank check, device program, device verify and device erase.

Device | Select/default... <F5>

This window allows selecting the desired type of the device from list of default devices. This one is a cyclic buffer in which are stored last 10 selected devices including its device options. This list is saved to disk by command **File | Exit and save**.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

Use a **<Del>** key for delete of current device from list of default devices. There isn't possible to empty this list, if you repeat this access. The last device stays in buffer and the **<Del>** key isn't accepted.

Device | Select device ... <Alt+F5>

This window allows selecting the desired type of the device from all devices supported by current programmer. It is possible to choose device by **name**, by **type** or by **manufacturer**.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device | Select/default...** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

Select device ... | All

This window allows selecting the desired type of the device from all devices supported by current programmer. Supported devices are displayed in a list box.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Select device ... | Only selected type

This window allows selecting the desired type of the device. At the first - you must select a device type (e.g. EPROM) and device subtype (e.g. 64Kx8 (27512)), using mouse or cursor keys. It will cause a list of manufacturers and devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Select device ... | Only selected manufacturer

This window allows selecting the desired device type by manufacturer. First select a required manufacturer in Manufacturer box using mouse or cursor keys. It will cause a list of selected manufacturer devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

Device | Select EPROM/Flash by ID... <Ctrl+F5>

Use this command for autoselect an EPROM or Flash as active device by reading the device ID. The programmer can automatically identify certain devices by the reading the manufacturer and the device-ID that are burnt into the chip. This only applies to EPROM or Flash that supports this feature. If the device does not support a chip ID and manufacturer's ID, a message will be displayed indicating this as an unknown or not supported device.

If more devices with identical chip ID and manufacturer's ID were detected, the list of these devices will be displayed. A corresponding device can be choosed from this list by selecting its number (or manufacturer name) from list and press **<Enter>** (or click **OK** button). Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

WARNING: *The control program only support this time EPROM's and Flash with 28 and 32 pins. Any of programmers determines pins number automatically. For other programmers you must enter this number manually.*

The programmer applies a high voltage to the appropriate pins on the socket. This is necessary to enable the system to read the device ID. Do not insert into the socket a device that is not an EPROM or Flash. It may be damaged when the programmer applies the high voltage.

We don't recommend apply this command to 2764 and 27128 EPROM types, because most of them ID not supports.

Device | Device options

Settings of this menu are used for programming process, serialization and associated file control. **Device | Device options | Edit device options** is always a device specific menu, please refer to device information window (pressing <Ctrl+F1>) to know its meaning.

Device | Device options | Operation options... <Alt+O>

All settings of this command are used for programming process control. This is a flexible environment which content items associated with current device and programmer type. Items, which are valid for the current device but aren't supported by current programmer, are disabled. These settings are saving to disk along with associated device by **File | Exit and save** command. Below are listed and briefly described all the operation options, divided into their categorical groups:

- group **Addresses**:
  - device start address (default 0)
  - device end address (default device size-1)
  - buffer start address (default 0)
  - split (none)
- group **Insertion test:**
  - insertion test (default ENABLE)
  - check ID bytes (default ENABLE)
- group **Command execution:**
  - erase before programming (default DISABLE)
  - blank check before programming (default DISABLE)
  - verify after reading (default ENABLE)
  - verify after programming (ONCE, TWICE)
  - verify options (nominal VCC)

Device | Device options | Serialization...

Serialization is special mode of program. When a serialization mode is activated, a specified value is automatically inserted on predefined address into buffer before programming each device. When more devices are programmed one by one, the serial number value is changed for each device automatically and inserted into buffer before programming device, so each device has unique serial number.
There are three types of serialization:

- Disabled
- Incremental mode
- From file mode

If a new device is selected, the serialization function is set to a default state i.e. disabled.
Actual serialization settings for actually selected device are saving to disk along with associated device by **File / Exit and save** command.
When incremental mode is active following actual settings are saved to configuration file: address, size, serial value, incremental step and settings of modes ASCII / BIN, DEC / HEX, LS byte / MS Byte first.
When from-file mode is active following actual settings are saved to configuration file: name of input serialization file and actual label, which indicates the line with actual serial number in input file.

Device | Device options | Serialization | Incremental mode

The **Incremental mode** enables to assign individual serial numbers to each programmed device. A starting number entered by user will be incremented by specified step for each device program operation and loaded in selected format to specified buffer address prior to programming of each device.
There are following settings, that user can modify for incremental mode:

S / N size

S / N size option defines the number of bytes of serial value which will be written to buffer. For Bin (binary) serialization modes values 1-4 are valid for S / N size and for ASCII serialization modes values 1-8 are valid for S / N size.

Address

Address option specifies the buffer address, where serial value has to be written. Note that

address range must be inside the device start and device end addresses. Address must be correctly specified so the last (highest or lowest) byte of serial value must be inside device start and device end address range.

Start value

Start value option specifies the initial value, from which serialization will start. Generally the max. value for serialization is $1FFFFFFF in 32 bit long word.

When the actual serial value exceeds maximum value, three most significant bits of serial number are set to zero. After this action the number is always inside 0..$1FFFFFFF interval (this is basic style of overflow handling).

Step

Step options specifies the increment step of serial value incrementation.

S / N mode

S / N mode option defines the form in which serial value has to be written to buffer. Two options are available:

- ASCII means the serial number is written to buffer as ASCII string. For example number $0528CD is in ASCII mode written to buffer as 30h 35h 32h 38h 43h 44h (‘0’ ‘5’ ‘2’ ‘8’ ‘C’ ‘D’), i.e. six bytes.
- Bin means the serial number is written directly to buffer. If the serial number has more than one byte length, it can be written in one of two possible byte orders. The byte order can be changed in „Save to buffer“ item.

Style

Style option defines serial number base. There are two options:

- Decimal numbers are entered and displayed using the characters ‘0’ through ‘9’.
- Hexadecimal numbers also use characters ‘A’ through ‘F’.

The special case is Binary Dec, that means BCD number style. BCD means the decimal number is stored in hexadecimal number, i.e. each nibble must have value from 0 to 9. Values A to F are not allowed as nibbles of BCD numbers.

Select the base in „Style“ options before entering numbers of serial start value and step.

Save to buffer

Save to buffer option specifies the serial value byte order to write to buffer. This option is used for Bin S / N mode (for ASCII mode it has no effect).

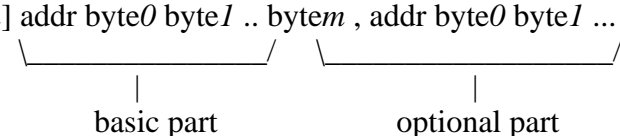
Two options are available:

- LSByte first (used by Intel processors) will place the Least Significant Byte of serial number to the lowest address in buffer.
- MSByte first (used by Motorola processors) will place the Most significant Byte first to the lowest address in buffer.

Device | Device options | Serialization | From file mode

Using the From file method, serial values are read from the user specified input file and written to buffer on address specified in input file.

This modality provides two settings:

File name

File name option specifies the file name from which serial addresses and values will be read. The input file for From file serialization must have special format, which is described in From file serialization file format below.

Start label

Start label defines the start label in input file. The reading of serial values from file starts from defined start label.

From file serialization input file includes addresses and arrays of bytes defining buffer addresses and data to write to buffer. Input file has text type format, which structure is:
[label*1*] addr byte*0* byte*1* .. byte*n*
...
[label*n*] addr byte*0* byte*1* .. byte*m* , addr byte*0* byte*1* ... byte*k*

```
          \_________________/   \___________________/
                   |                      |
               basic part           optional part
```

; Comment

meaning is:
basic part
Basic part defines buffer address and array of bytes to write to buffer. Basic part must be always defined after label in line.
optional part
Optional part defines the second array of bytes and buffer address to write to buffer. One optional part can be defined after basic part of data.
label*1*, label*n* - labels
Labels are identifiers for each line of input file. They are used for addressing each line of file. The labels should be unique. Addressing lines of file means, the required start label entered by user defines line in input file from which serial values reading starts.
addr -
Addr defines buffer address to write data following the address.
byte*0*..byte*n*, byte*0*..byte*m*, byte*0*..byte*k* -
Bytes arrays byte*0*..byte*n*, byte*0*..byte*m* and byte*0*..byte*k* are defining data, which are assigned to write to buffer. Maximum count of bytes in one data field following the address is 64 bytes. Data bytes are written to buffer from address addr to addr+n.
The process of writing particular bytes to buffer is:
byte*0* to addr
byte*1* to addr + *1*
byte*2* to addr + *2*
....
byte*n* to addr + *n*

Optional part is delimited from the first data part by character " , " (comma) and its structure is the same as in the first data part, i.e. address and following array of data bytes.

Characters with special use for labels:
[ ] - labels must be defined inside square brackets
',' – character which delimiters basic part and optional part of data
';' - the semicolon character means the beginning of a comment. All characters from „;„ to the end of line are ignored. Comment can be on individual line or in the end of definition line.

Note:
- *Label names can contain all characters except '[' and ']'. The label names are analysed as non case sensitive, i.e. character 'a' is same as 'A', 'b' is same as 'B' etc..*
- *All address and byte number values in input file are hexadecimal.*
- *Allowed address value size is from 1 to 4 bytes.*
- *Allowed size of data arrays in one line is in range from 1 to 64 bytes. When there are two data arrays in one line, the sum of their size in bytes can be maximally 80 bytes.*
- *Be careful to set correct addresses. Address must be defined inside device start and device end address range. In case of address out of range, warning window appears and serialization is set to disabled (None).*

Example:
[nav*1*] A7890 78 89 56 02 AB CD ; comment1
[nav*2*] A7890 02 02 04 06 08 0A
[nav*3*] A7890 08 09 0A 0B A0 C0 ; comment2
[nav*4*] A7890 68 87 50 02 0B 8D
[nav*5*] A7890 A8 88 59 02 AB 7D

;next line contains also second definition
[nav*6*] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16
; this is last line - end of file

In the example file six serial values with labels „nav*1*“, „nav*2*“, ...“nav*6*“ are defined. Each value is written to buffer on address $A7890. All values have size 6 bytes. The line with „nav*6*“ label has also second value definition, which is written to buffer on address $FFFF6 and has size 10 bytes, i.e. the last byte of this value will be written to address $FFFFF.

Device | Device options | Statistics...

Statistics gives the information about actual count of device operations, which were proceeded on selected type device. If one device is corresponding to one device operation, e.g. programming, the number of device operations will be equal to number of programmed devices.

The next function of statistics is **Count down**. Count down allows checking the number of device operations, and then number of devices, on which device operations have to be done. After each successful device operation the value of count down counter is decremented. Count down has user defined start number of devices to do. When count down value reach zero, it means, specified number of devices is complete and user message about complete count down will be displayed.
Statistics dialog contains following options:

Check boxes **Program**, **Verify**, **Blank**, **Erase** and **Read** define operations, after which statistics values increment.

Check box **Count down** sets Count down activity (enable or disable). Edit box following the Count down check box defines initial number of count down counter, from which count down starts.

Statistics dialog can be also opened by pressing right mouse button on Statistics panel and clicking diplayed item Statistics.

Actual statistics values are displaying in main window of control program in Statistics panel.
Statistics panel contains three statistics values – **Success**, **Failure**, **Total** and two **Count down** information values **Count down** and **Remains**. Meanings of these values are:

Success	number of operations which where successfully completed
Failure	number of operations which where not successfully completed
Total	number of all operations
Count down	informs about Count down activity (Enabled or Disabled)
Remains	informs about remaining number of device operations to do

Successful operation means any device operation of these, which is completed without errors:

- program
- verify
- blank check
- erase
- read

If device operation is finished with error(s) it is not successful operation.
When new device type is selected, all statistics values are set to zero and **Count down** is set to **Disabled**.
Reset button in **Statistics** panel reset statistics values.
Reload Count down button in **Statistics** panel reloads initial value to **Count down**.

Device | Device options | Associated file...
This command is used for setting associated file with current device. This is a file, which can be automatic loaded to buffer after device is selected from default devices select list or by start control program.
You can edit the associated file name in file name box, put a full pathname. The control program checks the present of this file on the disk. Also is possible enabling or disabling automatic load of this file.
You can save both settings i.e. associated file and enabling of automatic load of this file to disk by command **File | Exit and save**.

Device | Blank check <F6>

This command allows to blank check of all devices or its part if possible. The control program reports a result of this action by a write of a warning message to INFO window.

The menu command **Device | Device options | Operation options** allows to set another working area as the standard.

Device | Read <F7>

This command allows to read all device or its part into the buffer. The control program reports a finish of this action by write a message to INFO window.

The menu command **Device | Device options | Operation options** allows to set another working area as the standard. Setting an option Verify data after reading in this menu command means a higher reliability for device reading.

Device | Verify <F8>

This command compares the programmed data of the all device or its part with data in buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device | Device options | Operation options** allows to set another working area as the standard.

By the setting in the menu **Options | Display errors** the command lets to write the found errors on the display or write the found errors to VERIFY.ERR file. In the Display errors mode to the screen can display the program max. 45 the first found differences, which are located by the address where they were caused.

Device | Program <F9>

This command allows to programming of the all device or its part by the data of the buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device | Device options | Operation options** allows to set another working area as the standard, and set other operation options for programming process control.

Device | Erase <F10>

This command allows to erase the all programmable device. The program reports the end without error or end with the error by writes the warning report on the display.

Device | Test

This command executes a test with device selected from list of supported devices (e.g. static RAM) on programmers, which support this test.

Device | IC test

This command activates a test section for ICs separated by compatibility to any libraries (on distribution diskette). First select an appropriate library, wished device and then a mode for test vectors run (**Loop**, **Single step**). Control sequence and test results are displayed to Log window.

Device | JAM/VME/... Player

There are propietary devices made by the biggest manufacturers of integrated circuits of the world that allow special debugging operations on their products. Some programmers can integrate with these devices and manage them through the functions of this menu.
The specific descriptions of these special devices is out of the purpose of this manual.

Device | Device info... <Ctrl+F1>

The command provides additional information about the current device - size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

PROGRAMMER

The Programmer menu contains commands to communicate, diagnose status and use read and program functions of the programmer.

Options | Find programmer... <Ctrl+F>

Selects a new type of programmer and communication parameters. This command contains following items:

Programmer - sets a new type of programmer for find. If a Search all is selected, the control program finds all supported programmers.
Establish communication - allows manual or automatic establishing communication for a new programmer.
Speed - sets speed, if a manual establishing communication is selected, which PC sends data into the programmer. Speed is expressed as a percent from a maximal speed.
The communication speed modification is important for PCs with “slow” LPT ports, which haven’t sufficient driving power for a PC<->programmer cable (laptop, notebook, ...). Use this command, if you have any communication problems with connected programmer on the LPT port of your PC (e.g. control program reports a programmer absence, the communication with the programmer is unreliable, etc.).
If automatic establishing communication is selected, then control program sets a maximal communication speed.
Port - selects a communication port, which will be scanned for a requested programmer. If "All ports" is selected, the control program scans all the PC ports, which are available on standard addresses.
Address for special port - sets address of LPT port, if a Special port is selected.
Pressing key **<Enter>** or button **OK** initiates scanning for programmer by set parameters. There is same activity as at start the control program. The command clears a list of default devices without the current device, if the new selected programmer supports this one.

This setting is saved to disk by command **Options | Save options**.

Options | Refind programmer <Shift+Ctrl+F>

Repeats a programmer and communication parameters search operation already selected from previous menu. This command contains the items described for previous command.

Programmer | Handler...

In dialog **Handler** a Handler type and Handler communication parameters can be set. Handler is an external device for special control of device operations in control program. When None Handler is selected, this means default state of control program, i.e. device operations are controlled directly by user otherwise control program is in special mode, when device operations are controlled automatically with co-operation with Handler.
Dialog **Handler** contains following items:

Selected Handler select wished Handler type.
Search at port select a COM port, which will be scanned for a requested Handler.

Pressing key **<Enter>** or button **OK** initiates scanning for Handler by set parameters. If selected Handler type is **None**, no Handler scanning will be processed. Current Handler settings are saved to configuration file by command **Options | Save options** or when control program is closed.
Handler is not available for sale.

Programmer | Module options...

This option is used for multiple socket programmers for defining MASTER socket and activity of each socket. **MASTER socket** group box allows user to set socket which is preferentially used for device reading operation. **Enable/Disable socket** checkbox array allows user to set enabling and disabling of each socket individually. Disabled sockets are ignored for any device operation.

Programmer | Automatic YES!

This command is used for setting **Automatic YES!** mode. In this mode you just put a device into ZIF socket and a last operation will be repeated automatically. Program automatically detects an insertion of a new device and runs last executed operation without pressing any key or button. An insertion of device into ZIF is displayed on the screen. Repeated operation executing will be cancelled by pressing key **<Esc>** during waiting for insert/remove a device to/from ZIF.

Note: *During waiting for an insertion a new device into ZIF socket, the LED BUSY on the programmer is blinking.*

This mode may be enabled or disabled by item **Automatic YES!** mode. If a new programmer is selected **Options | Find programmer**, this mode will be disabled.

In **Response time** is possible to set a time interval within must be detected device in ZIF socket to accept an insertion of a new device. Default is set standard interval. If socket adapter is used then is recommended to set an elongated interval.

In **Pins with capacitors** bar may be entered a list of a pins interconnected by capacitors (for example: if a converter, which have connected capacitor between VCC and GND, is used), which may makes problems at detecting insertion of a new device.
List of pins of device is in form:
    pinA, pinB, pinC....
Example: 4,6,17
This list is erased if a new device is selected by **Device | Select/default** or **Device | Select device**

This setting is saved to disk by command **Options | Save options**.

Programmer | Selftest...

Command executes a selftest of current programmer without diagnostic POD. **ZIF SOCKET MUST BE EMPTY TO EXECUTE THIS COMMAND**. We recommend execute also **Diagnostics | Selftest plus** of programmer.

Programmer | Selftest plus...

Command executes a selftest of current programmer using diagnostic POD, which is included in standard delivery of programmer. We recommend run this test as often as possible, e.g. once every six months.

Programmer | Selftest ISP Connector...

Command executes a selftest of current programmer ISP connector using a specific diagnostic POD. Before the user supply this command he must execute the other two selftest commands and then he must follow the instructions of on line help.

Programmer | Calibration test...

Executes a test of calibration on many internal components and on used voltages.

OPTIONS

This menu let you view and change various default settings and operating modalities.

Options | General options...

General options dialog allows user to control following options of program.

File options

File options page allows you to set file masks, auto-reload of current file and choose file format recognizing for loaded files.

File format masks is used for setting file-name masks to use as a filter for file listing **in File | Save** and **File | Load file** window for all file formats. Mask must contain one of wildcards (*, ?) at least and must be applied correctly by syntax.

Project file default extension is used for setting project files-extension used as default extension in **File | Load project** and **File | Save project** dialogs.

In group **When current file is modified by another process** can be set mode of reloading of actually loaded (current) file. There are three choices:

1. Prompt before reloading file
2. Reload automatically
3. Ignore change scanning of current file

Load file format allows to set mode of file format recognition for loading files. When automatic file format is selected, program analyses format of loading file and test file for each of supported formats, that are available in program. If file format matches one of supported formats, the file is read to buffer in detected format.

Manual file format allows user to select explicitly wished file format from list of supported file formats. File may be loaded no completely or incorrectly, if file format does not match to user selected format.

Hex file options

This page contains several options for loading control by any of HEX formats.

The first option sets **erasing** buffer (with desired value) automatically before the loading by any of HEX formats.

The second option sets a **negative offset**, which is used for data addresses modification by loading from any HEX file so, that data can be written to existing buffer addresses.

Example:

A file contents data by Motorola S - format. A data block started at address FFFF0H. It is a S2 format with length of address array of 3 bytes. For all data reading you can set a value of negative offset to FFFF0H. It means, that the offset will be subtracted from current real addresses and so data will be written from buffer address 0.

Warning: *The value of negative offset is subtracted from real address and therefore a result of subtraction can be negative number. Because take care of correct setting of this value.*

Language

This page allows you to select another language for user interface such as menu, buttons, dialogs, information and messages. It also allows to select wished help file in another language. For another language support of user interface the language definition file is required.

Sound

Sound page allows user to select the sound mode of program. Program generates sounds after some activities, e.g. activities on device (programming, verifying, reading, etc.). Program generates sound also when warning or error message is displayed. User can now select sound from MS Windows system sound (required installed sound card), PC speaker or none sound.

Errors

This option allows you set a form of errors displaying as a result of programmed data verifying. Errors can be displayed to the **screen** (max. 45 differences), **saved to VERIFY.ERR** file on the disk in current directory or it will **not displayed**. In case the displaying errors are turned off, the control program reports a warning message in INFO window only.

Other

Page Other allows user to manage other program settings. In the panel Tool buttons, hint display options on toolbar buttons in main program window can be modified. In the panel Start-up directory can be selected mode of selecting directory when program starts. Default start-up directory means directory, from which program is called. Directory in which program was lastly ended means the last current directory when program was lastly ended. This directory assumes the first directory from directory history list.

Save options

Save options page allows you to select the program options saving when exiting program. Three options are available here:
- **Don't save options** during quiting program and don't ask for saving options.
- **Auto save options** during quiting program without asking for saving options.
- **Prompt for save options** before quting program. User can select to save or not to save options.

Options | View

Use the View menu commands to display or hide different elements of program environment such as toolbars.

Options | View | Main toolbar

Choose this command to show or hide the Main toolbar.

Options | View | Additional toolbar

Choose this command to show or hide the Additional toolbar.

Options | View | Programming conditions before programming

Choose this command to enable/disable display of Programming conditions before device program operation is confirmed.

Options | Protected mode...

Protected mode is special mode of program. When program is in Protected mode, there are disabled program operation and commands which can modify buffer or device settings. Protected mode is used for prevent operator from modify buffer or device settings due to insignificance. Protected mode is suitable for the programming of a large amount of the same type of devices. There are two ways how to switch program to Protected mode:

1. by using menu command **Options | Protected mode**. This command displays password dialog. User have to enter password twice to confirm the password is correct. After password confirmation program switches to Protected mode. The entered password is then used to switch off Protected mode.
2. by reading project, which was previously saved in Protected mode.

To switch program from Protected mode to normal mode use the menu command **Options | Normal mode**. The "Password required" dialog appears. User have to enter the same password as the password entered during switch to Protected mode.
Other way to cancel Protected mode of program is closing of program, because program Protected mode is active until program is closed. The next program start will be to Normal (standard) mode (the only exception is case of project loaded by command line parameter name of project and the project was saved in Protected mode).

Options | Save options

This command saves all settings that are currently supported for saving, even if auto-save is turned off. Following options are saved: options under the **Options** menu, ten last selected devices, file history, main program window position and size.

HELP

Pressing the **<F1>** key accesses the Help. When you selecting menu item and press **<F1>**, you access context sensitive help. If PG4UW is executing an operation with the programmer **<F1>** generates no response.

Since the Help system is continuously updated together with the control program, it may contain information not included in this manual.
Detailed information on individual menu commands can be found in the integrated on line Help.

Note: *Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult www.grifo.com or www.grifo.it.*

Help | Contents F1

This command shows all the on line help.

Help | Search for Help On...

This command shows the on line help page referenced to current window, command, box, data, value, format, etc. currently selected by cursor.

Help | Supported devices

This command displays list of all devices supported by at least one type of all supported programmers. It is usefull especially when user wants to find any device supported by at least one model of programmers.

Help | Supported programmers

This command displays information about programmers, where supported this program.

Help | Device list (current programmer)

This command makes a list of all devices supported by current programmer and saves its to **?????DEV.txt** text file and **?????DEV.htm** HTML file in the directory where control program is run from. Marks ????? are replaced by abbreviated name of current programmer, the device list is generated for.

Help | Device list (all programmers) <Shift+Alt+D>

This command makes a list of all devices supported by all **grifo®** programmers and saves its to **?????DEV.txt** text file and **?????DEV.htm** HTML file in the directory where control program is run from. Marks ????? are replaced by abbreviated name of current programmer, the device list is generated for.

Help | Device list (cross reference)

This command makes cross reference list of all devices supported by all programmers available on market and supported by this control program. The resulting list is in HTML format and consists of following files:

- one main HTML file **TOP_DEV.htm** with supported device manufacturers listed
- partial HTML files with list of supported devices for each device manufacturer

Main HTML file is placed to directory where this control program for programmers is located. Partial HTML files are placed to subdirectory **DEV_HTML** placed to the directory where control program for programmers is located.

Help | Create diagnostic report

This command is used for writing more particular diagnostic information to **Log window** and consequently copy **Log window** content to clipboard. The Log window content can be placed from clipboard to any text editor. Diagnostic report is useful when error occurs in control program or programmer and kind of the error is, that user can not resolve it oneself and he must contact programmer manufacturer. In this case when customer send message to manufacturer about his problem it is good to send also diagnostic report. Diagnostic report can help manufacturer to localise the reason of error and resolve it sooner.

Help | About...

This command shows a window where are reported information about current programmer, selected languages, general information on developer company and version number.

SOFTWARE ADDITIONAL FEATURES

PG4UW is common control program for all of the **grifo®** programmers. Thus, during work with him it is possible to find some items, not referenced to current selected programmer.

There is possible to attach to selected file "device associated file" too. This file may be automatically loaded into the buffer, if item **Device options |Associated file | Automatic load** is set to Yes. File can be associated manually or accepting last loaded file into the buffer. For this purpose is used item **Device options |Associated file | File name**.

There exist feature of control program, named **Automatic YES!**. In concrete, it mean there isn't necessary to press YES labeled button to repeat last activity, because auto sensing feature of programmer detect insertion of new chip into ZIF socked and start last executed automatically. This feature can be set in menu **Options**. If you use package converter with come pins coupled by capacitor, enter list of these pins.

The YES button allows to re-executes the last performed operation without any use of PC keyboard. in order to reduce the management time especially when numerous devices must be programmed. The location of YES button is described on figure 4.

Some special devices (e.g. Philips Coolrunner family) require external DAT files, that isn't present in standard PG4UW software delivery on mini CD. If you need to program these devices, look at **www.grifo.com** or **www.grifo.it**.

Please don't move any window when BUSY LED is on in fact the watching circuit can be activate to switch the programmer in safe status when communication with PC is slow or there are errors. For the same reason, during any operation on the target device it is preferible to reduce working load of PC (i.e. disable the contemporaneous execution of many other programs).

Take care to possibility of Windows operating system, to manages several applications in multitasking. There is needful for regular running of control program for any **grifo®** programmer that parallel port, on which is programmer connected, must be reserved for this program only. Otherwise, any other program must not simultaneously use (or any way modify) this port.

The control program can handle all modes of parallel ports (full IEEE 1284 support), thus you don't need to configure LPT port in special modality.

HOW TO START

This chapter contain all the information essential to connect the programmer to the PC, install the software and begin to use the programmer quickly. **Please read completely these steps before attempting any use of Your programmer**, especially when the operator has never used a programmer before. In order to satisfy the most frequent requests of the users, the chapter describes the steps necessary for making a copy of a device or program it with a file saved on PC.
For more detailed information about the technical features of each the and the control software, please refer to the specific chapters in previous pages of this manual.

a) Install the software PG4UW as described in CONTROL PROGRAM INSTALLATION paragraph. Whenever the program is not installed from mini CD, but from last version downloaded from web site, please follow the indications of NEW VERSION OF CONTROL PROGRAM paragraph.

b) Arrange the **UEP 50** as described in the PROGRAMMER ARRANGEMENT paragraph.

c) Connect the programmer to PC through the USB interface, as described in USB CONNECTION: DRIVER INSTALLATION paragraph. Aslternatively the programmer can be connected through the parallel interface, as described in LPT CONNECTION paragraph.

d) Execute the PG4UW program as described in RUN THE CONTROL PROGRAM paragraph.

e) Select the device You want to program, either through the **Device | Select device** menu or the keyboard shortcut <**Alt+F5**> (press key <**F5**> while keeping <**Alt**> key pressed) or more easily by a click on icon:

f) Load the data to program into the selected device:
- When data must be loaded from disk, use the **File | Load...** command or press the <**F3**> key, or more easily click the icon:

- When data must be taken by another device already programmed, first it must be read its content. So: open the ZIF socket (lever up), insert the source device taking care that the orientation coincide with the near design, close the ZIF socket (lever down) and perform the reading through the **Device | Read** command or press the <**F7**> key, or more easily click the icon:

At the end of reading, visualized either by control program and by GOOD LED of programmer, open the ZIF socket and remove the source device just read.

As the **UEP 50** programmer manages also microcontrollers, PLD and these can be protected from unauthorized reading, please verify if this protection is enabled (contact directly the device supplier) and if it is, please don't proceed with following steps.

g) Insert the device to program on the ZIF socket, by always taking care of orientation and position, and then close it.

h) Verify that the component to program is completely erased hrough the **Device | Blank check** command or press the <**F6**> key, or more easily click the icon:

i) Program the current content of buffer on the device hrough the **Device | Program** command or press the <**F9**> key, or more easily click the icon:

j) Now it is possible to compare the buffer content with the device just programmed hrough the **Device | Verify** command or press the <**F8**> key, or more easily click the icon:

k) Open the ZIF socket (lever up) and exctract the device just programmed. This last is ready to be mounted and tested on the customer's hardware.

WARRANTY TERMS

Warranty guarantees on failure free operating of the programmer and all its parts, materials and workmanship for **one years** from the date of purchase. This warranty is limited to 25,000 cycles on DIL ZIF socket or 10,000 cycles on non DIL ZIF sockets). If the product is diagnosed as defective, the authorized repair center will repair or replace defective parts at no charge. Parts used for replacement and/or whole programmer are warranted only for the reminder of the original warranty period.

The warranty does not apply to products that are of wear and tear or mechanically damaged. Equally, the warranty does not apply to products opened and/or repaired and/or altered by personnel not authorized by **grifo®**, or to products that have been misused, abused, accidentated or that were improperly installed.

Before sending the programmer, please contact **grifo®** directly to agree about the shipping modalities. However it is essential to return to **grifo®** the transport packing of the programmer in good state and complete of all the items (communication cable, power supply, diagnostic POD, adapters, etc.) and include a brief and exhaustive descrtiption of the problem encountered and a diagnostic report either in printed or electronic format.

For unwarrantable repairs You will be billed according to the costs of replacement materials, service time and freight. **grifo®** or its distributors will determine whether the defective product should be repaired or replaced and judge whether or not the warranty applies.

Please also see TROUBLESHOOTING chapter.

ADDITIONAL TOOLS

To extend easily the already large amount of devices programmable thorugh the **grifo®** programmers, the package converters allow to use also packaging types different from the classic DIL as the SDIP, PLCC, JLCC, SOIC, SOP, PSOP, SSOP, TSOP, TSOPII, TSSOP, QFP, PQFP, TQFP, VQFP, QFN (MLF), SON, BGA, EBGA, FBGA, VFBGA, UBGA, FTBGA, LAP, CSP, SCSP etc.
For more information about which package converter can be used with your programmer and device please use the information window shown by **Device | Device info...** (**<Ctrl+F1>**) command and then contact **grifo®** directly.
Please remind that programmer is compatible even with universal adapters or special adapters produced by third party companies. Also in this condition consult the information windos displayed by control program that advise about these possibility.

TROUBLESHOOTING

The programmer and used materials are high quality products and we really want you to enjoy our product. Nevertheless, problems can occur. In such cases please read carefully all the enclosed documentation again. Probably you will find the needed answer right away. When the problem persist, please follow the instructions below.
If You don't find the solution first it is necessary to create and transmit (by FAX or e-mail) the diagnostic report that shows the problem and then You can cotact directly **grifo®** by phone. Provide us an accurate description of Your PC configuration, including some other circumstances bearing on the problem in question. Don't forget please to tell PC type, manufacturer, speed, operating system type and version, resident programs; your communication port features.

READING OR PROGRAMMING PROBLEMS

- Assure the correct alignment of target device in the socket. Check the serigraph on the programmer or see the adapter serigraph if using one.

- Target device name must be selected on PG4UW before attempting to read or to program it.

- Target device name and selected device on PG4UW **MUST BE EXACTLY THE SAME!** If You attempted to program or to read a device with a name different or just similar to the target device name You may have damaged both the target device and the programmer.

- Update the control program. Electronic manifacturers create new devices almost everyday so it is essential to serch target device names in the latest version of PG4UW. If You still can't find Your target device please refer to paragraph UNSUPPORTED TARGET DEVICE.

- If the target device has never been programmed before or has been deleted it has returned to blank state. Perform Blank check test through **Device | Blank check** command or the shortcut <**F6**>.

- **UEP 50** can perform a self test to assure the correct working of its internal components. Please follow the instructions of DIAGNOSTIC SELFTEST paragraph.

- Some devices protect the content of their EPROM thorugh protection fuses. These fuses must be disengaged (using the specific options in **Device | Device Options | Operation options** menus). Otherwise programming or reading one of these devices results impossible, even if the device itself, the programmer and the PC are working correctly.

- When malfunction reading and programming are performed through ISP interface, check that user board is correctly powered. So please measure the power supply voltage with a tester directly on the selected and used device.

COMMUNICATION ERRORS

- Programmer must be supplied correctly, so the green POWER LED must be ON with high brightness and the power supply and the cable must be the ones delivered with the programmer. In order to test the power supply cable's physical integrity twist it delicately and see if something changes.

- Update the control program, often a more recent version includes enhancements also in communication reliability. Please refer to NEW VERSION OF CONTROL PROGRAM paragraph for more information.

- Programmer must be connected using USB cable from delivery.

- Try to install programmer and PG4UW on another PC. If your programmer works normally on the other computer you might have a problem with the first PC. Compare differences between both computers.

- Due a large variety of parallel port types, a case may occur when the programmer cannot establish communication with the PC. This problem may be shown either as none communication between the PC and the programmer, or by unreliable communication.

- If this behavior occurs, try to connect your programmer to other parallel ports or to some other PCs near you.

- Verify that the connected communication port is correctly present under Windows operating system, by using the proper window displayed under Start | Settings | Control panel | System properties | Hardware | Device manager | Ports (Com & LPT).

- Programmer must be connected directly to parallel port, no data switch, hardware keys or port duplicators can be put between PC and programmer.

- Parallel port must be completely dedicated to the programmer, under Windows it must not be shared as a network resource and no printer driver must use it.

- The parallel cable must be the one delivered with the programmer or a printer cable **with all the pins connected.** To test the cable's physical integrity twist it delicately when PG4UW tries to connect to the programmer and see if something changes.

- The parallel port where the programmer is connected **must be** configured as IEEE 1284 compatible. You should check in the BIOS configuration of Your computer that the port is set as ECP/EPP. When You use a multi I/O card, assure that it can be configured as IEEE 1284 compatible.

- When the best modality described in previous point still doesn't work correctlyt, try to set the parallel port as Centronics compatible (or standard, or normal, or SPP, according to the BIOS manufacturer: all these denominations refer to the same setting for the parallel port). This could decrease performances but ensures the highet communication reliability.

UNSUPPORTED TARGET DEVICE

If You need to use a target device not supported by control program, please follow the next steps:

- Look in the device list of the latest version of the control program on our Internet web sites. Your new target device might already be included in this version.

- Contact **grifo®** directly. We may need detailed data sheets of your target device and, if possible, samples. The samples will be returned to you after we include your target device in a new version of PG4UW.

APPENDIX A: ALPHABETICAL INDEX

A

ADDITIONAL TOOLS **57**
ASSISTANCE **1**
AUTOMATIC YES! **49**

B

BANANA JACK **11, 23**
BINARY **13**
BLANK CHECK **46**
BUFFER **3, 36**
BUSY **22**

C

CHECKSUM **38**
COMMANDS **33**
  BUFFER **36**
  DEVICE **39**
  FILE **33**
  HELP **52**
  OPTIONS **50**
  PROGRAMMER **47**
COMMUNICATION **59**
CONNECTORS **11, 26**
CONVERTERS **57**

D

DEVICE **3, 13**
DEVICE OPTIONS **41**
DEVICE SELECTION **39**
DIAGNOSTIC REPORT **53, 58**
DIMENSIONI **10**

E

ELECTROSTATIC NOISES **1**
ELEMENTS **11**
ENCRYPTION **35**
ERASE **47**
ERROR **51**
ERROR **22**

F

FILE FORMATS **13, 33**

G

GOOD **22**

H

HANDLER **48**
HELP **52**
HEX **3**, **13**
HOT KEYS **32**
HOW TO START **55**
HUMIDITY **8**

I

IC TEST **47**
IDE **50**
IEEE 1284 **59**
INCLUDED IN PACKAGE **8**
INSTALLATION **16**
ISP **10**, **24**, **50**

J

JEDEC **13**
JTAG **27**

L

LANGUAGE **51**
LEDS **11**, **18**, **22**
LPT **15**, **20**

M

MAIN WINDOW **31**
MOS **13**
MOTOROLA **13**
MULTIPROGRAMMING **22**
MULTITASKING **54**

O

OPERATIONS **12**

P

PACKAGE **57**
PARALLEL PORT **59**
PASSWORD **52**
PC CONFIGURATION **15**, **30**

PINDRIVERS **8**
POD **23**, **49**, **50**
POWER **8**, **22**
POWER SUPPLY **8**
PRINTER PORT **59**
PROBLEMS **58**
PROGRAM **47**
PROJECT **33**
PROJECT OPTIONS **35**
PROTECTED MODE **52**

R

READ **46**
REPORT **53**

S

SELFTEST **23**, **49**, **50**
SERIALIZATION **42**
SOFTWARE ADDITIONAL FEATURES **54**
SOFTWARE DESCRIPTION **30**
SOUND **51**
SPEED **10**
STATISTICS **45**

T

TEKTRONIX **13**
TEMPERATURE **8**
TERMINOLOGY **3**
TRADEMARKS **2**
TROUBLESHOOTING **58**

U

UPDATE **21**
USB **15**, **18**
USE **16**, **30**

V

VERIFY **46**
VERSION **21**

W

WARRANTY **2**, **57**
WEIGHT **8**

Y

YES! **11**, **49**

Z

ZIF **3**, **13**